Fachbereich III Informations- und Kommunikationswissenschaften

Institut für Angewandte Sprachwissenschaft

MAGISTERARBEIT

INTERNATIONALES INFORMATIONSMANAGEMENT

Web Content Mining nach Informationen zu wissenschaftlich tätigen Personen im Umfeld der Informationswissenschaft

vorgelegt von

Sarah Risse

Erstgutachterin: Prof. Dr. Christa Womser-Hacker

Zweitgutachter: Dr. Thomas Mandl

Hildesheim, im März 2006

Zusammenfassung

In der vorliegenden Arbeit wird ein Verfahren zur Suche nach Informationen zu Wissenschaftlern prototypisch für den Bereich der Informationswissenschaft entwickelt, in welchem Ansätze des Web Content Mining eingesetzt werden.
Zunächst werden Möglichkeiten und Probleme der Informationssuche im Web aufgezeigt, sowie verschiedene Verfahren des Web Content Mining beschrieben.
Das entwickelte Verfahren verwendet online Publikationsdienste und persönliche Homepages der Wissenschaftler als Quellen. Zur Suche in den Publikationsdiensten und der Informationsextraktion aus ihren Ergebnisseiten werden Wrapper konstruiert. Des Weiteren werden Methoden zur Informationsextraktion aus den Homepages implementiert, die auf Heuristiken zu Struktur und Aufbau der Seiten beruhen. Für die Suche nach persönlichen Homepages von Informationswissenschaftlern wird ein spezialisiertes Suchverfahren entwickelt.
Das Verfahren wird in einer Java-Applikation implementiert und anschließend evaluiert, um das Potenzial des gewählten Ansatzes zu untersuchen.

Schlüsselbegriffe:
Web Informationssuche, Web Content Mining Informationsextraktion, Wrapper, Spezialisierte Suchverfahren

Abstract

This thesis deals with the development of a search system for information on scientists which is implemented prototypically for the area of information science, employing Web Content Mining techniques.
Initially the field of web information search and its problems are characterized and Web Content Mining techniques are presented.
The sources that are used in the implemented approach are online publication services and personal homepages of scientists. Wrappers for querying the publication services and information extraction from their result pages are constructed, as well as methods for information extraction from the homepages, that are based on heuristics concerning structure and composition of the pages. Moreover a specialized search technique for searching for personal homepages of information scientists is developed.
The approach is implemented in a java application and finally evaluated to investigate its overall potential.

Keywords:
web information search, web content mining, web information extraction, wrapper, specialized search technique

Inhaltsverzeichnis

1 Einleitung

Die Bedeutung des Internets für die Wissenschaft hat im Laufe der Zeit stetig zugenommen, so dass es heute zu dem wohl wichtigsten Kommunikationsmedium und der Hauptinformationsquelle für Wissenschaftler avanciert ist. Sehr viele Wissenschaftler nutzen das World Wide Web (Web), um sich selbst und ihre Arbeit zu präsentieren, Digitale Bibliotheken bieten Informationen zu Publikationen und wissenschaftliche Veröffentlichungen sind häufig als Preprint bereits online verfügbar, bevor sie in gedruckter Form erhältlich sind.

Gleichzeitig hat aber auch das Problem der Informationsüberflutung mit dem rasanten Wachstum des Internets und seiner daraus resultierenden unüberschaubaren Größe an Bedeutung gewonnen: Es wird zunehmend aufwendiger, in der großen Menge von Daten relevante Informationen zu finden und so konkrete Informationsbedürfnisse zu befriedigen.

Möchte zum Beispiel ein Student mehr über die Forschungsaktivitäten und Publikationen eines Wissenschaftlers erfahren, von dem er eine interessante Veröffentlichung gelesen hat, oder möchte ein Wissenschaftler sich über einen möglichen Projektpartner informieren und sich unter anderem einen Überblick über dessen Veröffentlichungen, Projekte, Arbeitgeber und Kontaktdaten verschaffen, kann das Auffinden der für den Suchenden relevanten Informationen im Web ziemlich aufwendig sein.

Eine normale Suchmaschine wird für die Suche mit dem Namen der Person von Interesse in der Regel zwischen einigen hundert und mehreren tausend Seiten als Ergebnis liefern, von denen nur einige wenige relevante Informationen enthalten. Der Suchende muss zunächst sondieren, welche der referenzierten Seiten sich auf die gesuchte Person beziehen, denn mit großer Wahrscheinlichkeit haben mehrere Personen den gleichen Namen wie die gesuchte, und die Suchergebnisse beziehen sich dementsprechend auf jene unterschiedlichen Personen. Im weiteren Verlauf der Suche müsste der Nutzer die einzelnen Seiten aufrufen und aus der Menge der angegebenen Daten die für ihn relevanten Informationen herausfiltern. Für die Suche nach Publikationsangaben verwendet der Suchende vielleicht verschiedene online verfügbare Publikationsdienste. Dazu muss er zunächst für das Fach spezifische Quellsysteme finden und, um diese sinnvoll zu nutzen, mit den Suchmasken und Ausgabeformaten der verschiedenen Dienste vertraut sein. Der Suchende vergleicht vielleicht die Ergebnisse mehrerer Pub-

likationsdienste, um eine möglichst vollständige Übersicht über die Veröffentlichungen der gesuchten Person zu erhalten.

Ein System zur Suche nach Informationen zu Wissenschaftlern, das die für den Suchenden irrelevanten Informationen außer Acht lässt und ihm quasi ‘auf einen Blick’ die relevanten Informationen liefert, würde dem Suchenden die Befriedigung seines Informationsbedürfnisses wesentlich erleichtern. Dazu müsste das System relevante Informationsquellen konsultieren, die gewünschten Informationen extrahieren und diese dem Nutzer in integrierter Form präsentieren.

Der Bereich des Web Content Mining umfasst derartige Ansätze zur Verbesserung der Informationssuche und -nutzung im Web. In diesem Forschungsbereich werden u.a. Verfahren zur Extraktion und Integration von Informationen aus Webdokumenten und -datenbanken, zur Optimierung von Suchmaschinenergebnissen sowie der Entwicklung spezialisierter Suchverfahren zusammengefasst.

Im Rahmen der vorliegenden Arbeit soll mit Methoden des Web Content Mining ein Verfahren zur Suche nach Informationen zu Wissenschaftlern prototypisch für den Bereich der Informationswissenschaften entwickelt werden. Dazu sollen geeignete Quellen ausgewählt und Verfahren zur Informationsextraktion aus diesen erstellt werden. Die extrahierten Informationen sollen dem Suchenden in integrierter Form präsentiert werden. Das Verfahren wird in einer Java-Applikation implementiert. Abschließend soll eine Evaluierung des Prototypen zeigen, wie Erfolg versprechend der gewählte Ansatz ist.

In Kapitel 2 werden daher als Grundlage Möglichkeiten und Probleme der Informationssuche im Internet beschrieben und Verfahren und Anwendungsbereiche des Web Content Mining charakterisiert, die zum Ziel haben, die Informationssuche im Web zu verbessern.

Daraufhin wird in Kapitel 3 der auf dieser Basis entwickelte Prototyp vorgestellt. Dabei werden zunächst die Zielsetzung eingegrenzt und die Rahmenbedingungen benannt. Die Auswahl der in dem Suchverfahren verwendeten Quellen wird begründet und die Informationsextraktion aus diesen beschrieben. Das für den Prototyp entwickelte Verfahren zur spezialisierten Suche nach persönlichen Homepages von Informationswissen-

schaftlern wird vorgestellt und schließlich ein prozessorientierter Systemüberblick gegeben.

Kapitel 4 behandelt die Implementierung in Form einer Java-Applikation. Die Benutzeroberfläche wird beschrieben und die einzelnen Module des Programms erläutert.

Die anschließende Evaluierung wird mit der verwendete Vorgehensweise und den erzielten Ergebnissen in Kapitel 5 vorgestellt. Darauf aufbauend werden Verbesserungsmöglichkeiten für das System aufgezeigt. Den Abschluss bilden eine Zusammenfassung und eine kritische Betrachtung der Arbeit.

2 Informationssuche im Web

Im Folgenden werden zunächst einige der im Kontext des entwickelten Prototypen relevanten Eigenschaften des World Wide Web (Web) beschrieben. Darauf aufbauend werden aktuelle Möglichkeiten der Informationssuche im Web mit Ihren Defiziten thematisiert. Web Content Mining versucht unter anderem, einige dieser Defizite zu beseitigen und die Möglichkeiten der Informationssuche und -erschließung zu verbessern.

2.1 Eigenschaften des Web

Brin und Page [1998, 112] beschreiben das Web als „vast collection of completely uncontrolled heterogeneous documents“ und fassen damit die wichtigsten Eigenschaften zusammen.

Das Web ist eine Sammlung von miteinander verknüpften Hypertextdokumenten, deren Größe auf im zehnfachen Terrabyte Bereich liegend geschätzt wird.

Die Heterogenität der Webdaten betrifft verschiedene Aspekte. So sind u.a. die vorhandenen Daten in Format und Art unterschiedlich (z.B. Texte, Bilder, HTML-Seiten etc.), die verwendeten Sprachen, der Zweck der Nutzung (privat, wissenschaftlich, kommerziell), die Zugänglichkeit (z.B. öffentlich oder passwortgeschützt) und die Arten der Erstellung (manuell vs. automatisch) [vgl. Hoff 2002, 32f.] sind ebenfalls sehr divers.

Darüberhinaus besteht das Web aus dem so genannten *Surface Web* und dem *Deep Web* [vgl. Liu, Chang 2004, 1]. Das Deep Web, auch *Invisible Web* genannt, besteht aus den Webseiten, die von Suchmaschinen bewusst oder aus technischen Gründen nicht erfasst werden [vgl. Lewandowski 2005b, 51]. Von Sherman und Price wird dies folgendermaßen definiert:

> „Text pages, files or other often high-quality authoritative information available via the World Wide Web that general-purpose search engines cannot, due to technical limitations, or will not, due to deliberate choice, add to their indices of Web pages.”
>
> [Sherman und Price 2001, 57]

Das Surface Web hingegen beinhaltet die statischen, untereinander verlinkten Seiten, zu denen der Nutzer browsen kann [vgl. Liu, Chang 2004, 1]. Untersuchungen gehen davon aus, dass das Deep Web ca. 500 Mal größer als das Surface Web und zudem wesentlich reicher an qualitativ hochwertiger Information ist [vgl. Bergman 2001].

Ein weiterer im Hinblick auf die maschinelle Erschließung und Aufbereitung der Daten relevanter Aspekt ist der Strukturiertheitsgrad der Webdokumente.

Generell werden Textdokumente aus Sicht der Informationsverarbeitung in freie, strukturierte und semistrukturierte Texte unterschieden. Mit freiem Text sind natürlichsprachliche Texte wie z.B. Zeitungsartikel gemeint, die grammatikalisch korrekte und vollständige Sätze enthalten [vgl. Eikvil 1999, 8]. Strukturierter Text ist definiert als „textual information in a database or file following a predefined and strict format“ [Eikvil 1999, 8]. Semistrukturierter Text ist weder freier noch strukturierter Text. Er folgt keinen festen Formatregeln, ist oft ungrammatikalisch und schlagwortartig verfasst [vgl. Eikvil 1999, 9].

Basierend darauf, dass alle Webseiten auf Grund ihrer visuellen Aufbereitung über eine gewisse Struktur verfügen, werden sie oft pauschal als semistrukturiert klassifiziert [vgl. Eikvil 1999, 10]. Diese Struktur lässt sich meist aus der Verwendung der HTML-Tags ableiten. Hinzu kommt, dass in Webseiten oft keine ganzen Sätze verwendet werden, sondern die Information schlagwortartig präsentiert wird. Webseiten können aber auch strukturiert oder völlig unstrukturiert sein. Im Folgenden werden dementsprechend Webdokumente, abhängig davon wie ihr Inhalt organisiert ist, als strukturiert, semistrukturiert oder unstrukturiert bezeichnet. Webseiten, die Informationen aus zu Grunde liegenden Datenbanken mit Hilfe von Templates darstellen, wie z.B. die Ergebnisseiten von Suchmaschinen oder Online-Shops, sind in der Regel strukturiert, während manuell erstellte Seiten eher weniger strukturiert sind [vgl. Eikvil 1999, 10f.]. Es ist zu berücksichtigen, dass der Strukturiertheitsgrad immer von den betrachteten Attributen abhängig ist [vgl. Hsu et al. 1998, 535]. So kann eine strukturierte Webseite, wie z.B. die Webseite von *Spiegel-Online*[1], unstrukturierte Daten enthalten, in diesem Fall die als freier Text vorliegenden einzelnen Berichte.

Darüber hinaus enthalten einzelne Webseiten oft eine Mischung aus unterschiedlichsten Informationen, wie z.B. Hauptinhalt, Werbeanzeigen und Navigationsleisten [vgl. Liu, Chang 2004, 2]. Der einzelne Nutzer kann die Informationen in für ihn nützliche und überflüssige unterschieden. Für eine Anwendung aber, die auf den Webdaten aufsetzt, ist diese Unterscheidung nicht offensichtlich, so dass man in dieser Hinsicht von verrauschten Daten spricht.

[1] http://www.spiegel.de (verifiziert am 29.03.2006)

Die vorgestellten Eigenschaften des Web erschweren die Arbeit existenter Suchdienste, die den Nutzer beim Auffinden von bestimmten Informationen unterstützen, und beeinflussen deren Ergebnisqualität in Bezug auf die Befriedigung des Informationsbedürfnisses des Suchenden. Im nächsten Abschnitt sollen diese Suchdienste mit ihren u.a. auf den vorgestellten Eigenschaften des Web basierenden Problemen dargestellt werden.

2.2 Tools zur Unterstützung der Informationssuche

Die am häufigsten zur Unterstützung der Informationssuche im Internet genutzten Tools sind Web-Verzeichnisse und Suchmaschinen.

Web-Verzeichnisse bieten dem Nutzer manuell erstellte, thematisch sortierte Sammlungen von Web-Dokumenten, die hierarchisch strukturiert sind. Um zu einem relevanten Dokument zu gelangen, muss der Nutzer durch die vorgegebenen Kategorien browsen. Der zur Erstellung eines Web-Verzeichnisses benötigte manuelle Aufwand kann jedoch der Größe des Web nicht gerecht werden, so dass nur ein Bruchteil der vorhandenen Dokumente in diesen Verzeichnissen aufgeführt werden kann. Hinzu kommt, dass es auf Grund der hohen Dynamik des Web schwer ist, ein solches Verzeichnis aktuell zu halten. Ein weiterer Nachteil dieser Verzeichnisse ist die Subjektivität des Hierarchiesystems und der entsprechenden Einteilung der Dokumente. Möglicherweise hat ein Nutzer eine andere Vorstellung von dieser, so dass er Dokumente nicht findet, obwohl sie vorhanden sind [vgl. Hoff 2002, 71f.].

Suchmaschinen sind die am häufigsten von Nutzern zur Informationssuche verwendeten Tools. Der Suchende formuliert sein Informationsbedürfnis in Form von Stichwörtern die mit booleschen Operatoren verknüpft werden können, und das System liefert als Ergebnis eine nach Relevanz sortierte Liste von Web-Dokumenten, die die jeweiligen Suchterme enthalten. Bei dieser Art der Suche treten Probleme auf, die allen Information Retrieval Systemen gemein sind, wie z.B. das Vokabelproblem bei Homonymen und Synonymen [vgl. Hoff 2002, 55]. Darüber hinaus entstehen aber auch andere Probleme, die zum Teil aus den oben genannten Eigenschaften des Web resultieren.

So leiden Suchmaschinen unter einer niedrigen Precision und einem niedrigen Recall[2]. Zum einen werden viele nicht-relevante Ergebnisse zurückgeliefert, so dass es aufwen-

[2] Recall und Precision sind Standard-Evaluierungsmaße für Informationretrievalsysteme. Siehe dazu z.B.: Salton, G.; McGill, M.(1987): Information Retrieval - Grundlegendes für Informationswissenschaftler. Hamburg et al.: Mc Graw Hill.

dig ist, die relevanten Informationen zu finden. Zum anderen können Suchmaschinen nur einen kleinen Teil des Web indexieren, so dass nicht-indexierte Informationen nicht gefunden werden können. Die Ursachen dafür liegen sowohl in der unmöglich komplett zu erfassenden Menge an Dokumenten als auch darin, dass Dokumente des Deep Web den Suchmaschinen nur teilweise oder gar nicht zugänglich sind, wie in Kapitel 2.1 erläutert wurde.

Ein weiteres Problem der Suchmaschinen ist die Aktualität der Ergebnisse. Die Menge der Dokumente ändert sich ständig, so dass Untersuchungen zufolge Ergebnislisten häufig seit bis zu vier Monaten veraltet sind [vgl. Bergman, 2001]. Auch dies hängt mit der Größe des Web zusammen, die dazu führt, dass Zeit vergeht, bis neue Seiten indexiert werden.

Über diese, eher datenbezogenen Probleme hinaus ergeben sich aus den eingeschränkten Möglichkeiten, die dem Informationssuchenden zur Interaktion mit einer Suchmaschine geboten werden, weitere: Es ist für den Suchenden schwer, ein spezifisches Informationsbedürfnis über das generische Eingabefeld einer Suchmaschine auszudrücken, denn diese lassen „nur“ Anfragen in der Form verknüpfter Stichwörter zu [vgl. Cohen et al. 2000, 19]. Dazu kommt, dass die Suchmaschine nicht über Wissen bezüglich des Bezugsrahmens einer Anfrage verfügt und diesen Kontext auch nicht aus einer Anfrage, die in der Regel aus einzelnen Wörtern besteht, ableiten kann. Damit ist es für den Nutzer noch schwerer, eindeutige Anfragen zu spezifizieren [vgl. Hoff 2002, 1].

Suchmaschinen liefern dem Suchenden in der Regel lediglich eine Liste mit Kurzbeschreibungen der gefundenen Webseiten und Verweise zu diesen. Sucht der Nutzer aber ein bestimmtes Informationsobjekt[3], und nicht eine Webseite zu seiner Anfrage, muss er einzeln die Ergebnisseiten durchgehen, um zu prüfen, ob sie die gewünschte Information enthalten. Oft werden auch zusammenhängende Informationen gesucht, die über verschiedene Seiten verteilt sind. Derartige Informationsbedürfnisse zu befriedigen ist für den Suchenden mit hohem Aufwand verbunden und Suchmaschinen können in diesen Fällen meist nicht genügend Hilfestellung bieten [vgl. Laender et al. 2002, 1].

[3] Mit dem Begriff Informationsobjekt soll die Information bezeichnet werden, die der Benutzer benötigt, um sein aktuelles Informationsbedürfnis zu befriedigen. Was genau dieses Informationsobjekt ist, ergibt sich also aus dem Suchkontext des Nutzers und könnte z.B. die E-Mail-Adresse oder Telefonnummer einer Person sein oder der Kaufpreis eines Buches. Im Englischen spricht man von *item of data* oder *information item*. Im weiteren Verlauf wird der Begriff Informationsitem synonym mit dem Begriff Informationsobjekt verwendet.

Um die Informationssuche im Web mit ihren hier beschriebenen Problemen zu verbessern und den Informationsreichtum des Web erschöpfender nutzen zu können, sind verschiedenste Verfahren entwickelt worden, die zu dem Bereich des Web Content Mining zusammengefasst werden können [vgl. Chen, Chue 2005, 1227], welcher im nächsten Kapitel beschrieben wird.

2.3 Web Mining und Web Content Mining

Web Content Mining ist ein Unterbereich des Web Mining. Daher wird zunächst der Bereich des Web Mining beschrieben und anschließend genauer auf Methoden und Anwendungsbereiche des Web Content Mining eingegangen.

2.3.1 Web Mining

Etzioni [1996] bezeichnet mit Web Mining die Anwendung von Data Mining Techniken auf Webdaten mit dem Ziel, automatisch Informationen in Webdokumenten und -services zu entdecken und zu extrahieren. Diese Definition ist in vielen darauf folgenden Forschungsberichten zu finden [u.a. bei Kosala, Blockeel 2000; Chen, Chue 2005; Chau et al. 2003] und wird in diesen weiterentwickelt: Berendt et al. [2002, 266] definieren Web Mining als „the application of data mining techniques to the content, structure, and usage of Web resources“. Sie zeigen damit schon die verschiedenen Aspekte des Web auf, die in der Forschung von Interesse sind. Diese Definitionen sehen die Verwendung von Data Mining Techniken als notwendige Voraussetzung dafür an, um von Web Mining sprechen zu können, und sind somit sehr eng.

Bei anderen Autoren wird Web Mining als „discovery and analysis of useful information from Web data“ definiert [Chakrabarti 2003; Cooley et al. 1997; Liu, Chang 2004, 1; Madria et al. 1999, 1]. Im Kontext dieser Definition ist Data Mining neben Maschinellem Lernen, *Natural Language Processing* (NLP), Statistik, Datenbanken und Information Retrieval lediglich ein Forschungsbereich aus dem Methoden und Techniken verwendet werden [vgl. Liu, Chang 2004, 1]. Diese Sichtweise soll auch dieser Arbeit zu Grunde gelegt werden.

Das Web umfasst drei verschiedene Arten von Daten: Nämlich „[content] data on the Web, the web log data regarding the users who browsed the web pages and the web structure data“ [Madria et al. 1999, 1]. Dementsprechend werden, je nach Art der unter-

suchten Daten, drei Teilbereiche des Web Mining unterschieden: Web Structure Mining, Web Usage Mining und Web Content Mining [vgl. u.a. Chau et al. 2002, 2; Liu, Chang 2004, 1; Kosala, Blockeel 2000, 3].

Web Structure Mining befasst sich mit der Analyse der Hyperlink-Strukturen im Web mit dem Ziel, Informationen über Beziehungen und Ähnlichkeiten verschiedener Seiten zu erhalten. Eines der wichtigsten Anwendungsgebiete ist die Qualitäts- bzw. Relevanzbewertung von Webseiten. Ausgehend von der Annahme, dass ein Link von einer Webseite auf eine andere für die Qualität dieser verlinkten Seite spricht, wird angenommen, dass die Relevanz einer Seite mit der Anzahl Links, die von anderen Seiten auf sie verweisen, zunimmt. Das prominenteste Anwendungsbeispiel dafür ist der *PageRank*-Algorithmus von Google [vgl. Brin, Page 1998].

Im Zentrum des Interesses des Web Usage Mining steht die Interaktion des Benutzers mit dem Web. Die Daten, die dazu untersucht werden, sind Protokolle der Anfragen, die Benutzer an eine Website gestellt haben, die in den so genannten Log-Dateien der Server gespeichert werden [vgl. Berendt et al. 2002, 267]. Analysen können so z.B. zeigen, welche URL-Pfade von Benutzern häufig verwendet werden, oder welche Seiten häufig zusammen konsultiert werden [vgl. Madria et al. 1999, 13]. Daraus resultierende Vorhersagen des Benutzerverhaltens finden unter anderem Anwendung in E-Commerce-Applikationen oder bei der Personalisierung von Webangeboten [z.B. Cooley et al. 2000].

Web Content Mining schließlich befasst sich mit der Nutzung des Web als Informationsressource und versucht, bestehende Nutzungsmöglichkeiten zu verbessern sowie neue Wege der Erschließung zu entwickeln. Die Forschung in diesem Bereich beinhaltet das Auffinden von Informationsquellen, die Analyse bzw. Aufbereitung der Retrievalergebnisse und die Informationsextraktion aus Webseiten [vgl. Chen, Chue 2005, 1226]. Diese Teilaufgaben müssen nicht zwingend in einer Anwendung integriert sein, sondern können auch Inhalt eigenständiger Anwendungen sein. Im nächsten Abschnitt werden Verfahren und Anwendungsbereiche des Web Content Mining näher erläutert.

2.3.2 Methoden und Anwendungsbereiche des Web Content Mining

2.3.2.1 Optimierung der Suchmaschinenergebnisse

Ein Anwendungsbereich des Web Content Mining ist die Verbesserung der Nutzung und der Ergebnisse von Suchmaschinen [vgl. Chen, Chue 2005, 1227]. Dies kann zum einen während des Indexierungsvorgangs geschehen, oder aber durch Post-Retrieval Analyse.

Focused Crawling

Eine Optimierung der Suchmaschinenergebnisse während es Indexierungsvorgangs kann durch Methoden des Focused Crawling [Chakrabarti et al. 1999] erreicht werden. Eine mögliche Methode ist hierbei die Verwendung eines Crawlers, der nur Seiten indexiert, die mit einem festgelegten Thema übereinstimmen, indem er die Relevanz von gefundenen Links und Seiten über Vergleiche mit vorher festgelegten Beispieldokumenten bestimmt.

> „A focused crawler starts at a set of representative pages on a given topic and forces the crawler to stay focused on this topic while gathering web pages. A topic is defined with the help of a hypertext classifier which is pre-trained with a representative data set and corresponding topical classes".
>
> [Klopotek 2003, 64]

Mit dieser Form der Erstellung eines spezialisierten Index kann die Qualität der Suchergebnisse insofern verbessert werden, dass zu themenspezifischen Anfragen mehr relevante und weniger nicht-relevante Dokumente zurückgegeben werden. Zudem kann solch ein spezialisierter Index auf Grund seines geringeren Umfangs mit weniger Aufwand aktuell gehalten werden [vgl. Steele 2001, 1].

Post-Retrieval-Analyse

Verfahren der Post-Retrieval-Analyse dienen der aufbereiteten Darstellung von Suchergebnissen. Dazu werden Klassifikations- und Clusteringtechniken[4] angewandt, um ähnliche Suchergebnisse zusammenzufassen oder Kategorien zuzuordnen, so dass der Benutzer die Menge der Retrievalergebnisse besser handhaben kann. Die Suchmaschine NorthernLight[5] z.B. ordnet die Suchergebnisse vordefinierten Kategorien zu. Vivisimo[6] hingegen fasst die Suchergebnisse in *on-the-fly* generierten Clustern zusammen. Liu und Chang [2004, 2] bezeichnen diese Verfahren der Post-Retrieval-Analyse als „Building Concept Hierarchies".

Spezialisierte Suchmaschinen

Die beschriebenen Verfahren kommen zum Teil in spezialisierten Suchmaschinen in integrierter Art und Weise zum Einsatz. „Spezialisierte Suchmaschinen schränken ihre Suche auf eine bestimmte Menge von Webdokumenten ein" [Hoff 2002, 57]. Diese Einschränkung kann nach unterschiedlichen Kriterien erfolgen; denkbar sind u.a. regionale oder sprachliche Kriterien, oder aber die Einschränkung auf ein bestimmtes Thema („topic") oder eine bestimmte Kategorie („category") von Webseiten, wie z.B. Homepages oder Blogs [Hoff 2002, 57; Steele 2001, 1]. Durch eine solche Spezialisierung fallen einige der beschriebenen Probleme herkömmlicher Suchmaschinen wie die mangelnde Aktualität der Ergebnisse weg oder sind weniger stark ausgeprägt; dadurch, dass nur ein Bruchteil des Web relevant ist, ist es wesentlich leichter, die Ergebnismenge aktuell zu halten. Durch die thematische Festlegung ist der Bezugsrahmen einer Anwendung bekannt. Außerdem können die Eingabemöglichkeiten und angebotenen Suchfelder dem entsprechenden Thema angepasst werden [vgl. Steele 2001, 1]. Handelt es sich z.B. um eine spezialisierte Suchmaschine für Filme, wären Suchfelder zur Eingrenzung der Suche auf Filme eines bestimmten Regisseurs oder Schauspielers denkbar.

[4] Klassifikations- und Clusteringverfahren sind klassische Bereiche des Maschinellen Lernens. „A clustering algorithm discovers groups in the set of documents such that documents within a group are more similar than documents across groups. A classifier is first trained with a corpus of documents that are labelled with topics. Later the classifier is presented with unlabeled instances and is required to estimate their topics reliably." [Chakrabarti 2003, 8f.]

[5] http.//www.northernlight.com (verifiziert am 29.03.2006)

[6] http://www.vivisimo.com (verifiziert am 29.03.2006)

2.3.2.2 Informationsextraktion und -integration

Ein weiterer Bereich des Web Content Mining befasst sich mit der Informationsextraktion aus Webseiten. Wie in Kapitel 2.1 beschrieben, können Webseiten unstrukturierte, semistrukturierte oder strukturierte Daten enthalten. Dementsprechend variieren je nach Datentyp die Extraktionsmethoden. Im Hinblick auf die nötigen Extraktionsmethoden kann der Strukturiertheitsgrad von Webseiten noch differenzierter beschrieben werden:

> „ […] a Web page that provides itemized information [can be considered] as structured if each attribute in a tuple can be correctly extracted based on some uniform clues, such as delimiters or the orders of the attributes. On the other hand, a Web page is unstructured if linguistic knowledge is required to extract the attributes correctly. […] Semistructured Web pages are those that are not unstructured. Semistructured Web pages may contain tuples with missing attributes, attributes with multiple values, variant attribute permutations, exceptions and typos."
>
> [Hsu et al. 1998, 535f.]

Informationsextraktion (IE) beschäftigt sich allgemein damit, gezielt bestimmte Informationen in Texten zu identifizieren und zu extrahieren. Anders formuliert ist das Ziel von IE, „to transform text into a structured format and thereby reducing the information in a document to a tabular structure"[Eikvil 1999, 3].

IE ist ursprünglich ein Forschungsfeld der NLP und befasst sich traditionell mit der Extraktion von Informationen aus natürlichsprachlichen Dokumenten. IE Systeme sind nicht darauf ausgerichtet, den Text der zu bearbeitenden Dokumente zu verstehen, sondern analysieren die Teilbereiche der Dokumente, die relevante Information enthalten. Was relevante Information sind, wird durch die vorab festgelegte Aufgabenstellung bestimmt, die angibt, welche Informationen das System finden soll [vgl. Eikvil 1999, 3].

Kern eines jeden IE-Systems sind die Regeln, die beschreiben, wie die gesuchte Information extrahiert werden soll (engl. *extraction patterns* oder *rules*) [Muslea 1999, 1]. Die Art dieser Regeln ist abhängig von der Beschaffenheit der Texte, aus denen die Information extrahiert wird.

Informationsextraktion aus unstrukturierten Webseiten

IE-Systeme zur Handhabung von freiem Text nutzen Methoden aus dem Bereich der NLP. Die Extraktionsregeln basieren u.a. auf syntaktischen Analysen, semantischen Informationen und Eigennamenerkennung. Zur Extraktion von unstrukturierten Webdaten werden diese Techniken ebenfalls verwendet, oft in Kombination mit Verfahren des

Maschinellen Lernens [vgl. Liu, Chang 2004, 2]. Darüber hinaus gibt es neuere Forschungen, die von *Common Language Patterns* (üblicherweise verwendete Satzstrukturen, mit denen bestimmte Fakten oder Beziehungen ausgedrückt werden) und der Datenredundanz im Web Gebrauch machen, um Konzepte und Eigennamen und deren Beziehungen zu finden [vgl. Liu, Chang 2004, 2]. Eine weitere Richtung dieser Forschung ist die Nutzung des Web als Korpus für *Question Answering* [vgl. Liu, Chang 2004, 2].

Informationsextraktion aus (semi-)strukturierten Webseiten

Für IE aus semistrukturierten bzw. strukturierten Webseiten sind die NLP-Techniken traditioneller IE-Systeme nicht geeignet, da oft keine ausreichend grammatikalische Struktur vorhanden ist [vgl. Eikvil 1999, 9]. Hinzukommt, dass NLP-Techniken relativ langsam sind, was in Kombination mit den großen Mengen an Dokumenten und der häufigen Anforderung der Extraktion zu Anfragezeiten ein Problem darstellt.

Wie bereits in Kapitel 1.1 angesprochen, ist die Struktur von Webseiten in der Regel durch die graphische Aufbereitung (Schriftarten und -farben, Tabellen etc.) gegeben und für den Nutzer explizit erkennbar. Um diese Layout-Aspekte umzusetzen, werden oft ursprünglich zu Strukturierungszwecken gedachte HTML-Tags „missbraucht“. IE-Systeme versuchen diese Strukturierung aus der Verwendung der HTML-Tags und erkennbaren Regelmäßigkeiten abzuleiten. Zusätzlich ist bei der IE aus Webseiten die Hyperlinkorganisation des Web zu berücksichtigen. Häufig ist es notwendig, Hyperlinks zu verfolgen, um die gesamte gewünschte Information zu erhalten [vgl. Eikvil 1999, 10].

IE-Systeme, die Daten aus strukturierten und semistrukturierten Webseiten mittels site-spezifischer Extraktionsregeln extrahieren, werden *Wrapper*[7] genannt. Diese Extraktionsregeln basieren auf Mustern, die die Begrenzung der zu extrahierenden Bereiche widerspiegeln, und nutzen keine linguistischen Eigenschaften. Man spricht von Delimiter-basierten Extraktionsregeln [vgl. Muslea 1999, 3].

Wrapper können sowohl manuell, als auch automatisch oder semi-automatisch erstellt werden.

[7] Der Term Wrapper kommt ursprünglich aus dem Datenbank Umfeld und bezeichnet dort “a software component that converts data and queries from one model to another“ [Eikvil 1999, 12]

In ersterem Fall wird aufbauend auf der Analyse der Struktur der Website manuell ein Extraktionsprogramm erstellt [vgl. Eikvil 1999, 14], wobei zur Unterstützung spezielle Sprachen, mit denen sich die Struktur der Webseiten leichter beschreiben lässt, eingesetzt werden können [vgl. Hsu et al. 1998, 536]. Der Nachteil der manuellen Erstellung von Wrappern liegt darin, dass sie nicht auf andere Sites übertragen werden können und somit für jede Domain neu angepasst werden müssen [vgl. Eikvil 1999, 14]. Aus diesem Manko heraus haben sich semi-automatische und automatische Methoden der Wrappergenerierung entwickelt.

Bei der semi-automatischen Wrappererstellung werden zur Unterstützung des Wrapperdesignprozesses Tools mit graphischen Schnittstellen bereitgestellt, die dem Nutzer ermöglichen, die Informationen, die extrahiert werden sollen, in einigen Beispieldokumenten zu markieren. Das Tool generiert dann den Programmcode anhand dieser markierten Beispiele [vgl. Eikvil 1999, 14].

Zur automatischen Erstellung von Wrappern werden Techniken des Maschinellen Lernens angewandt. Ausgehend von einer Menge manuell ausgezeichneter Trainingsdaten lernt das System die notwendigen Extraktionsregeln. [vgl. Eikvil 1999, 14].

Neben diesen Verfahren gibt es Versuche, die Wrappergenerierung für sehr strukturierte Seiten komplett zu automatisieren. In diese Kategorie fallen Seiten, deren Inhalte Einträge aus Datenbanken sind, die durch Templates formatiert dargestellt werden. Automatisierte Verfahren versuchen, ohne menschliches Eingreifen Muster in diesen Seiten zu finden und basierend auf diesen Mustern Extraktionsregeln zu generieren [vgl. Liu, Chang 2004, 2].

Web Informationsintegration

Informationsintegration im Web befasst sich mit der integrierten Nutzung verschiedener Webquellen. Dabei handelt es sich in der Regel um Datenbanken des Deep Web, auf deren Daten über ein Webinterface zugegriffen wird.

Ein Bestandteil ist die Fusion der aus den Ergebnisseiten dieser Datenbanken extrahierten Informationen, wobei das Problem der semantischen Heterogenität der Daten zu lösen ist.

> „The problem is that information might be organized in different ways with different vocabularies. So, an integration system needs to either learn or have access to semantic descriptions of these sources. This can be done either by bundling semantic information with Web pages or learning ontologies from the sources."

[Kambhampati, Knoblock 2003, 15]

Aufbauend auf der Beschreibung der einzelnen Quellen werden die Daten zusammengeführt. Dabei müssen die Einträge aus den verschiedenen Quellen, die sich auf dieselbe Entität beziehen aber in unterschiedlicher Struktur vorliegen, identifiziert werden. Diese Aufgabe wird als *Name Matching* oder *Object Matching* bezeichnet [vgl. Kambhampati, Knoblock 2003, 15]. Die dazu verwendeten Verfahren beinhalten sowohl regel-basierte Methoden, bei denen die Konditionen, unter den zwei Einträge gleich sind, manuell spezifiziert werden, als auch probabilistische Methoden, die Maschinelle Lernverfahren einsetzen [vgl. Bilenko et al. 2003, 17].

Mit der Integration der extrahierten Daten geht die Suche in den verschiedenen Quellsystemen einher. Jedes Quellsystem hat seine eigene Suchsyntax, doch der Nutzer stellt seine Anfrage in einem zentralen Query-Interface. Um den Besonderheiten der einzelnen Systeme gerecht zu werden und ihr Potenzial voll auszuschöpfen, muss die Nutzeranfrage in Suchanfragen für die einzelnen Quellsysteme übersetzt werden [vgl. Kambhampati, Knoblock 2003, 15]. Liu und Chan [2004, 2] bezeichnen diesen Bereich als *Web Query Interface Integration*.

2.3.2.3 Webseiten-Segmentierung

Webseiten bestehen typischerweise aus verschiedenen Bereichen, wie dem Hauptinhaltsbereich, der Navigationsleiste, gegebenenfalls Werbebereichen etc. Die Forschung in dem Bereich „Segmenting Web Pages and Noise Detecting" befasst sich mit dem automatischen Erkennen und Separieren dieser verschiedenen Bereiche [vgl. Liu, Chang

2004, 2]. Diese Methoden können unterstützend für andere Bereiche des Web Content Mining eingesetzt werden: So hat sich gezeigt, dass Kategorisierungs- und Clustering-Verfahren bessere Ergebnisse erzielen, wenn vorab nicht-inhaltstragende Bereiche wie Werbung oder Navigationsbereiche, entfernt wurden [vgl. Liu, Chang 2004, 2]. Auch Informationsextraktionsverfahren können bessere Leistungen erzielen, wenn der Hauptinhaltsbereich einer Seite bekannt ist. Zudem kann die Indexierung der Webdokumente verbessert werden, wenn die Auswahl der Indexterme anhand des inhaltstragenden Bereichs vorgenommen wird (vgl. Lewandowski 2005b, 221 ff.).

Ein anderer Anwendungsbereich dieser Forschung ist die Nutzung des Web von einem Gerät mit einem kleinen Display, wie z.B. einem Handy. Wenn die verschiedenen Inhaltsbereiche identifiziert werden können, können die Seitenlayouts so verändert werden, dass die Inhaltsbereiche ohne Informationsverlust auf einem kleinen Display dargestellt werden können [vgl. Liu, Chang 2004, 2].

3 Der entwickelte Prototyp

In diesem Kapitel wird das im Rahmen dieser Arbeit entwickelte Suchsystem vorgestellt und die der Umsetzung zugrunde liegenden Entscheidungen beschrieben. Die konkrete Implementierung des Prototypen hingegen wird im nächsten Kapitel behandelt. Zunächst wird die Aufgabe des Prototypen eingegrenzt und die Rahmenbedingungen benannt sowie ein Überblick über die generelle Vorgehensweise gegeben. Im zweiten Teil werden die zur Erfüllung des Suchziels verwendeten Quellen vorgestellt und die Auswahl dieser begründet. Die Erschließung der verschiedenen Quellen und die Extraktion der relevanten Informationen aus diesen werden erläutert. Abschließend wird in einem prozessorientierten Systemüberblick das Zusammenspiel der einzelnen Komponenten zusammengefasst.

3.1 Aufgabenstellung und Rahmenbedingungen

3.1.1 Suchziel und Suchsituation

Wie in der Einleitung beschrieben, sollen die Realisierungsmöglichkeiten für die spezialisierte Suche nach Informationen zu aktiven Wissenschaftlern am Beispiel der Informationswissenschaften aufgezeigt werden. Unter welchen Bedingungen und mit welchen Mitteln das Verfahren auf andere Fachbereiche portierbar ist, wird im Ausblick beleuchtet.

Dieser Aufgabe entsprechend ist das Ziel des entwickelten Suchverfahrens dem Suchenden zu ermöglichen, sich einen ersten Überblick über im Umfeld der Informationswissenschaft wissenschaftlich tätige Personen, ihre Forschungsaktivitäten und Publikationen zu machen. Als dafür interessante und relevante Informationen wurden folgende Daten identifiziert:

- der akademische Titel
- Kontaktdaten, zunächst begrenzt auf die Email-Adresse
- Einrichtung, für die die Person aktuell tätig ist
- ein Foto der Person
- Liste der Publikationen
- Lebenslauf
- Projekte, an denen die Person beteiligt ist/war

Das dem Nutzer präsentierte Suchergebnis soll nicht wie bei herkömmlichen, allgemeinen Suchmaschinen eine Liste von Referenzen zu Webseiten sein, von denen angenommen wird, dass sie die gesuchten Informationen enthalten. Vielmehr sollen die einzelnen Informationsitems aus den Quell-Webseiten extrahiert und dem Nutzer in einer übersichtlichen Zusammenstellung präsentiert werden.

Der Anfrageterm ist dem Informationsbedürfnis entsprechend, der Name der Person, zu der die Informationen gesucht werden, bestehend aus Vor- und Nachname. Im Folgenden wird die Person, zu der die Informationen gesucht werden, als Zielperson bezeichnet. Es werden Vorname und Nachname als Suchterme verwendet, um das Risiko zu verringern, dass Namensambiguitäten auftreten. Es wird angenommen, dass die Anfragen durch das Hinzunehmen des Vornamens der Zielperson eindeutiger werden.

In der ersten Entwicklung sollen nur Wissenschaftler aus dem deutschsprachigen Raum berücksichtigt werden. So wird versucht, zusätzliche Komplexität durch Mehrsprachigkeit und den damit im Regelfall verbundenen Einsatz von Spracherkennungs- und Übersetzungswerkzeugen zu vermeiden. Im Ausblick wird eine multilinguale Erweiterung des Systems skizziert.

Weiterhin soll die Suche in den Quellsystemen sowie die Extraktion und Integration der Informationen zur Anfragezeit durchgeführt werden und nicht auf einem vorab erstellten Index basieren, wie es bei Suchmaschinen normalerweise der Fall ist.

Zusammenfassend ist die Aufgabe des Suchverfahrens also, auf eine Anfrage in Form des Vor- und Nachnamens eines im deutschsprachigen Raum tätigen Informationswissenschaftlers als Ergebnis die oben aufgeführten Attribute, falls vorhanden, aufzufinden und dem Nutzer in aggregierter Form zu präsentieren. Falls die gesuchten Informationen nicht vorhanden, bzw. mit den angewandten Methoden und verwendeten Quellen nicht auffindbar sind, soll das System dies erkennen und dem Nutzer mitteilen.

3.1.3 Generelle Vorgehensweise

Nach der Festlegung der Suchaufgabe, der zu findenden Informationen und der Rahmenbedingungen wurde zunächst eine manuelle Suche der Zielinformationen für eine Gruppe von 14 Testpersonen durchgeführt, um einen Überblick über mögliche Quellen und einen ersten Eindruck zu eventuellen Schwierigkeiten zu erhalten. Diese Informationswissenschaftler wurden der Gruppe der Autoren, deren Aufsätze in dem Sammel-

band *Eibl, M.; Wolff, Ch.; Womser-Hacker, Ch. (Hrsg.) (2005): Designing Information Systems. Festschrift für Jürgen Krause zum 60. Geburtstag. UVK: Konstanz* veröffentlicht sind, entnommen[8]. Diese Testmenge wurde auch im weiteren Verlauf als Referenzmenge zur Erstellung der in den Extraktionsregeln verwendeten Heuristiken verwendet und zudem zur fortlaufenden Evaluierung während der Entwicklung eingesetzt.

Im nächsten Schritt wurden verschiedene Quellen, die Informationen zu Publikationsdaten von Wissenschaftlern anbieten, näher betrachtet und hinsichtlich der Erschließungs- und Extraktionsmöglichkeiten, sowie der Datenqualität vergleichend analysiert. Die Digitale Bibliothek CiteSeer und der Bibliographie-Server DBLP wurden als Quellen ausgewählt (siehe Kapitel 3.2.1). Als wichtigste Quelle für die anderen Informationsitems (s.o.) wurde die persönlichen Homepage der jeweiligen Zielperson ausgewählt (siehe Kapitel 3.2.2, 3.2.3). Anschließend wurden die Extraktionsregeln für die einzelnen Quellen und ein Verfahren zur spezialisierten Suche nach persönlichen Homepages von Informationswissenschaftlern entwickelt. Im letzten Schritt wurde eine grafische Oberfläche zur Interaktion mit dem Nutzer erstellt.

Anmerkung zur Effizienz

In Hinblick auf die Nutzerzufriedenheit spielt es eine große Rolle, die Suchdauer möglichst kurz zu halten. Mit Suchdauer wird hier die Zeit bezeichnet, die zwischen dem Stellen der Anfrage durch den Nutzer und der Anzeige der Suchergebnisse liegt. Vor dem Hintergrund, dass die Suche und die Extraktion der Informationen zur Anfragezeit durchgeführt werden, wird davon ausgegangen, dass vor allem die Anzahl der Zugriffe auf Webseiten und das damit einhergehende Einlesen der Seiten die Suchdauer beeinflussen. Folglich wurde versucht, die Anzahl der Webseitenaufrufe möglichst gering zu halten. Prinzipiell stand die Effizienz für die durchgeführte Exploration möglicher einzusetzender Quellen und Verfahren allerdings nicht im Vordergrund der Einscheidungen.

[8] Die Trainingsmenge sowie die in ihr ermittelten Eigenschaften sind im Anhang aufgeführt.

3.2 Quellen: Auswahl und Erschließung

Auf Basis der manuell durchgeführten Suchen und des Erfahrungswissens wurde entschieden, jeweils die persönliche Homepage der Zielperson als Hauptquelle zu verwenden. Es hat sich gezeigt, dass diese mit großer Wahrscheinlichkeit die gesuchten Informationsitems enthält. Von Vorteil bei der Verwendung der Homepage als Quelle ist zudem, dass die enthaltenen Informationen in der Regel eindeutig der Person zuzuordnen sind, die durch die Homepage vorgestellt wird.

Zusätzlich werden Bibliographie-Verzeichnisse und Digitale Bibliotheken neben den in der Regel auf den Homepages zu findenden Publikationslisten als Quellen für die Angaben zu den Veröffentlichungen der Person hinzugezogen. Ein Grund dafür ist, dass ein Eintrag in solchen Systemen als Indikator für die Bedeutung einer Publikation bzw. eines Autors gesehen wird, wobei Bekanntheitsgrad und Qualität nicht unbedingt gleichzusetzen sind, da die Aufnahme von Publikationen in die Systeme nicht immer einer Qualitätsprüfung unterliegt. Darüber hinaus beinhalten einige Digitale Bibliotheken Angaben darüber, wie oft und von welchen Veröffentlichungen ein bestimmter Artikel zitiert wurde. Diese Zitationsanalyse wird häufig verwendet, um die wissenschaftliche Bedeutung einzelner Werke zu messen [vgl. Goodrum et al. 2001, 662]. Es ist also wünschenswert, diese Zitationsangaben zusammen mit der Liste der Veröffentlichungen als Ergebnis zu erhalten.

3.2.1 Auswahl und Erschließung der Quellen für Publikationsdaten

Im folgenden Abschnitt werden die als Quellsysteme für die Publikationsdaten in Frage kommenden Systeme *CiteSeer*, *DBLP*, *DAFFODIL* und *Google Scholar* vorgestellt und verglichen. Die Entscheidung, DBLP und CiteSeer als Quellsysteme zu verwenden, wird begründet, und anschließend die Umsetzung der Extraktion und Integration der Publikationsdaten aus den beiden Systemen beschrieben.

3.2.1.1 Suchdienste für Publikationen

CiteSeer

CiteSeer[9] ist eine Digitale Bibliothek, die auf wissenschaftliche Literatur aus dem Bereich der Informatik und der Informationswissenschaften spezialisiert ist[10]. Das System wurde am *NEC Research Institute*[11] entwickelt. CiteSeer bietet einen Online-Suchindex für wissenschaftliche Veröffentlichungen, die frei im Web zugänglich sind [vgl. Goodrum et al. 2001, 664]. Eine Besonderheit von CiteSeer ist, dass zusätzlich zu Artikeln auch Zitationen in den Index aufgenommen werden und durchsuchbar sind. Zu jedem Artikel wird aufgeführt, wie viele und welche anderen Veröffentlichungen diesen Artikel zitiert haben und die Zitationsangaben sind mit den entsprechenden Artikeln verlinkt, falls sie in dem Index vorhanden sind [vgl. Lawrence et al. 1999, 67f.]. So entsteht ein Netz aus Artikeln, die über bidirektionale Zitationsbeziehungen miteinander verknüpft sind.

Zur Erstellung des CiteSeer-Index werden Suchmaschinen und Focused Crawler verwendet. Mit Hilfe der Suchmaschinen werden geeignete Ausgangspunkte für das Crawlen nach wissenschaftlichen Veröffentlichungen, die als PDF- oder Postscript-Dokument vorliegen, lokalisiert [vgl. Lawrence et al. 1999, 68]. Dabei gilt als Indikator für eine wissenschaftliche Veröffentlichung das Vorhandensein eines Bibliographieteils [vgl. Goodrum et al. 2001, 664]. Neben dieser automatischen Erschließung können Autoren ihre Veröffentlichungen auch zur Aufnahme in den Index anmelden [vgl. Lawrence et al. 1999, 68].

Auf den Index kann über eine Schlagwortsuche zugegriffen werden, wobei nach Artikeln oder Zitationen unterschieden werden kann. Bei der Suche nach Artikeln kann die

[9] http://citeseer.ist.psu.edu (verifiziert am 29.03.2006)
[10] vgl. http://citeseer.ist.psu.edu/citeseer.html (verifiziert am 28.03.2006)
[11] http://www.nec-labs.com (verifiziert am 29.03.2006)

Anfrage auf den Titel oder den Header der Dokumente, der u.a. Titel, Autorenangaben und Abstract enthält, eingeschränkt werden, bei der Suche nach Zitationen dagegen auf den Titel und Autorennamen [vgl. Lawrence et al. 1999, 70; Petinot et al. 2004, 557]. Per default wird zunächst eine boolesche Suche durchgeführt, in der die einzelnen Bestandteile einer Anfrage, falls nicht anders spezifiziert, mit *AND* verknüpft werden. Führt dies zu keinem Ergebnis wird diese Verknüpfung automatisch in eine *OR*-Verknüpfung umgewandelt, d. h. es werden auch Veröffentlichungen als relevant bewertet, die nur einen der Bestandteile des Anfrageterms enthalten.

Die Suchergebnisse werden in einem zweistufigen System präsentiert. In der Ergebnisliste für die Suche nach Artikeln ist jeder Treffer mit Titel, Autorennamen und Veröffentlichungsjahr, der Anzahl der Zitationen, falls vorhanden, und der URL der Fundstelle angegeben. Da der Platz begrenzt ist, werden oft nur der Anfang des Titels und nur ein Teil der Autorennamen aufgeführt. Die vollständigen Angaben sind in den einzelnen Veröffentlichungsseiten zu finden; diese enthalten jeweils zu einer Publikation die kompletten Angaben zu den Autoren und dem Publikationskontext, die Links zu den Volltexten, einen Ausschnitt aus den in dem Artikel zitierten Veröffentlichungen und eine Liste der Zitationen dieses Artikels in anderen Veröffentlichungen. Jeder Treffereintrag in der Ergebnisliste ist mit der zugehörigen Veröffentlichungsseite verlinkt.

Zur Aktualität und Vollständigkeit des Indexes konnten keine Angaben oder Untersuchungen gefunden werden.

DBLP

Der DBLP-Server[12] ist ein Bibliographie-Server für Informatik-Fachinformationen [vgl. Ley 1997, 1; 7], der seit 1993 an der Universität Trier betrieben wird. Ursprünglich auf die Themen Datenbanksysteme und Logikprogrammierung beschränkt, enthält der DBLP-Server heute auch u.a. Bibliographie-Informationen zu den Bereichen Algorithmen, Künstliche Intelligenz, Bioinformatik, Compilerbau, Kryptologie, Digitale Bibliotheken, Verteilte Systeme, Hypertext, Information Retrieval, Maschinelles Lernen und Multimedia[13]. Der manuell erstellte Index beinhaltet Einträge zu Beiträgen aus Tagungsbänden, Zeitschriften und Veröffentlichungsreihen und Büchern zu den genannten Fachbereichen.

[12] http://www.informatik.uni-trier.de/~ley/db/ (verifiziert am 29.03.2006)
[13] http://www.informatik.uni-trier.de/~ley/db/subjects.html (verifiziert am 29.03.2006)

Das Netz aus Publikationsmedien, Einzelpublikationen und Autoren ist über Hyperlinks verknüpft und über verschiedene Ansichten zugänglich. Eine davon ist die Anzeige von Autorenseiten, die für jeden Autor generiert werden.

> „Eine Autorenseite zählt alle dem System bekannten Publikationen der betreffenden Person auf. Jedes Vorkommen eines Autorennamens außerhalb „seiner" Seite ist durch Hyperlinks mit der betreffenden Autorenseite verknüpft"
>
> [Ley 1997, 4].

Die einzelnen Publikationseinträge bestehen aus den Namen der Autoren, dem Titel und dem Publikationskontext [vgl. Ley 2002, 7] und sind nach Aktualität sortiert. Andere Sichten auf den Datenbestand sind die Inhaltsverzeichnisse der Zeitschriften und Tagungsbände, von denen aus man zu den Autorenseiten gelangen kann [vgl. Ley 1997, 4].Neben diesem browsingorientierten Zugriff kann nach Autoren oder Titelstichwörtern gesucht werden und eine fortgeschrittene Suche ermöglicht die kombinierte Suche über die einzelnen Bestandteile der Publikationseinträge (Autoren, Titel, Veröffentlichungsort, Seitenzahlen, Jahr der Veröffentlichung).

Es sind keine offiziellen Angaben zur Aktualität des Indexes der DBLP zu finden, es scheint jedoch, dass Publikationen relativ zeitnah zum Veröffentlichungszeitpunkt in den Index aufgenommen werden.

Google Scholar

Google bietet mit Google Scholar[14] eine auf wissenschaftliche Inhalte spezialisierte Suche an[15]. Im Gegensatz zu z. B. CiteSeer wird keine thematische Einschränkung vorgenommen, sondern alle Wissenschaftsbereiche sollen abgedeckt werden [vgl. Lewandowski 2004, 1]. Neben Aufsätzen aus Zeitschriften und Tagungsbänden beinhaltet der Suchindex auch Bücher, Preprints, Abstracts und studentische Arbeiten[16]. Als Quellen werden sowohl frei zugängliche Websites, wie die von Universitäten und anderen wissenschaftlichen Einrichtungen, als auch passwortgeschützte Inhalte von Wissenschaftsverlagen und Fachgesellschaften verwendet [vgl. Lewandowski 2004, 1; Lewandowski 2005a, 17]. Über die erfassten Quellen, den Umfang und die Aktualität des Dokumentenbestands werden keine näheren Angaben gemacht. Erste Untersuchungen haben aber

[14] http://scholar.google.com/ (verifiziert am 29.03.2006)
[15] Google Scholar ist seit November 2004 online und bislang handelt es sich noch um eine Beta-Version. Die im Folgenden beschriebenen Mängel sind also vor dem Hintergrund des frühen Entwicklungsstadium zu betrachten.
[16] vgl. http://scholar.google.com/scholar/about.html (verifiziert am 29.03.2006)

ergeben, dass Vollständigkeit und Aktualität des Indexes nicht zufrieden stellend sind [vgl. Mayer, Walter 2005].

Alle Dokumente werden im Volltext erschlossen. Zudem werden Metadaten wie Autor, Titel, Zeitschriftennamen, Publikationsjahr etc. erfasst. Dementsprechend ist sowohl eine Schlagwortsuche im Volltext möglich, aber auch eine Einschränkung auf einen oder mehrere Autorennamen oder die Kombination verschiedener Metadatenfelder. Die Ergebnisse werden in einer nach Relevanz gerankten Trefferliste ausgegeben. Zu jedem Treffer werden „[…] der Titel, die keywords in context [!], die Autoren, die Quelle (Zeitschriften oder Webseite) sowie die Anzahl der innerhalb des Google-Datenbestandes ermittelten Zitationen angegeben“ [Lewandowski 2004, 1]. Dem Rankingverfahren liegt das in der Websuche verwendete Verfahren zu Grunde, wobei die Gewichtungen an den wissenschaftlichen Kontext angepasst wurden. Das genaue Verfahren ist dabei ebenso wie das in der „normalen Websuche“ von Google verwendete Verfahren unbekannt [vgl. Lewandowski 2005a, 13].

Neben der mangelnden Aktualität und der Unvollständigkeit des Indexes wurden zudem Fehler in der Extraktion von Autorennamen beobachtet, so dass die Zuordnungen von Autoren zu Publikationen häufig fehlerhaft sind [vgl. Lewandowski 2005a, 21f.]. Außerdem enthält der Datenbestand relativ viele Dubletten, so dass auch die Zitationsangaben nicht korrekt sind [vgl. Lewandowski 2005a, 22].

DAFFODIL

DAFFODIL[17] ist ein Projekt der Arbeitsgruppe Informationssysteme an der Universität Duisburg: Es ist „ein System zur integrierten Suche in heterogenen Digitalen Bibliotheken eines Fachgebiets unter Zusammenführung der Ergebnisse“ [Klas et al. 2005, 1]. Man kann es auch als „Virtuelle Digitale Bibliothek“ [Fuhr et al. 2000, 247] bezeichnen. Dieser Suchansatz wurde prototypisch für den Bereich der Informatik realisiert. Über eine einheitliche Suchmaske wird die Suche in mehr als zehn Digitalen Bibliotheken und weiteren Informationsquellen - u.a. CiteSeer, ACM Digital Library, DBLP und Achilles - ermöglicht [vgl. Klas et al. 2005, 1]. Dazu leiten Agenten und Wrapper die Suchanfragen an die Informationsanbieter weiter und führen die gefundenen Ergebnisse zusammen [Klas et al. 2005, 3]. Duplikate werden eliminiert [Schaefer 2005, 23] und das Ergebnis dem Benutzer „in Form einer gewichteten Resultatliste präsentiert“ [Klas

[17] http://www.daffodil.de (verifiziert am 29.03.2006)

et al. 2005, 3]. Die Treffereinträge in der Ergebnisliste enthalten jeweils den oder die Autorennamen sowie den Titel und das Jahr der Veröffentlichung. Die einzelnen Einträge sind mit Publikationsseiten verlinkt, die ausführlichere Informationen zu der entsprechenden Veröffentlichung enthalten, ähnlich wie bei CiteSeer.

Es kann sowohl eine Volltextsuche durchgeführt werden, als auch eine Suche über die Metadaten (Titel. Autor, Veröffentlichungsjahr). Zusätzlich kann die Auswahl der benutzten Quellen und der Typ der gesuchten Dokumente (PDF, PS etc.) eingeschränkt werden.
Der Prototyp wird zur Nutzung als Web-Applikation bereitgestellt. Es besteht aber auch die Möglichkeit, seine Funktionalität als Webservice zu nutzen.

3.2.1.2 Auswahl und Informationsextraktion

Folgende Kriterien wurden bei der Auswahl der Quellen berücksichtigt: Zum einen die Vollständigkeit und Aktualität der Datenbestände und zum anderen der möglichst geringe, zur Einbindung der Quelle benötigte Aufwand. Zusätzlich ist es von Bedeutung, dass die Strukturiertheit der Ergebnisdarstellung zuverlässige Extraktionsregeln erlaubt.

Betrachtet man vor diesem Hintergrund die in Frage kommenden Quellsysteme, so ist DAFFODIL die am besten geeignete Quelle, da die Vollständigkeit der Ergebnisse durch die integrierte Suche in verschiedenen Datenbeständen sehr gut ist. Dadurch dass alle anderen vorgestellten Quellsysteme in DAFFODIL integriert werden, ist die Vollständigkeit der Ergebnisse mindestens genauso gut und wahrscheinlich besser als die der anderen Systeme. Darüber hinaus enthält DAFFODIL die weiter oben angesprochen Angaben zur Zitationsanalyse, die für die Qualitätseinschätzung einzelner Artikel und die wissenschaftliche Bedeutung einer Person wichtig sind. Im Fall der Nutzung von DAFFODIL entfällt zudem die Entwicklung von Methoden zur Extraktion und Integration der heterogenen Information verschiedener Einzelsysteme, wodurch eine enorme Aufwandsersparnis erzielt werden kann.

Jedoch war zum Zeitpunkt der Entwicklung die Möglichkeit der Einbindung der DAFFODIL-Funktionen über Web Services nicht bekannt und es konnte auch keine alternative Möglichkeit der Einbindung ermittelt werden. Die Einbindung von DAFFODIL in den realisierten Prototypen wird aber in Kapitel 5.3 näher beschrieben.

Gegen die Verwendung von Google Scholar sprechen vor allem die Mängel in der Qualität und Aktualität der Ergebnisse, die wahrscheinlich auf das frühe Entwicklungsstadium dieses Dienstes zurückzuführen sind. Zudem ist bei der Verwendung von Google Scholar problematisch, dass der Suchindex nicht themenspezifisch ist, sondern versucht, den gesamten Bereich wissenschaftlicher Literatur abzudecken. Es ist anzunehmen, dass in einem allgemeinen Index die Wahrscheinlichkeit, dass ein Name nicht eindeutig ist, zunimmt. Es müsste also zusätzlich eine Möglichkeit der Namensdisambiguierung gefunden werden. Von der Verwendung von Google Scholar als Quellsystem wird daher abgesehen.

In Folge dessen wurden der DBLP-Server und CiteSeer als Quellsysteme ausgewählt. Der DBLP-Server enthält eine umfangreiche Sammlung von Veröffentlichungen aus Konferenzbänden, Schriftenreihen und Zeitschriften, unter denen auch die aus Sicht der Informationswissenschaft Relevanten vertreten sind (z.B. *ISI* oder *Mensch und Computer*). Die Publikationen eines Autors werden in Form einer klar strukturierten HTML-Seite ausgegeben, was die Erstellung eines Wrappers für diese Quelle erheblich erleichtert. Zur Vollständigkeit und Aktualität der CiteSeer Inhalte liegen keine Angaben vor. Ein Grund für die Auswahl von CiteSeer ist aber, dass zu jeder Publikation die Anzahl der Referenzen in anderen Veröffentlichungen angegeben wird, was wie oben beschrieben aus Sicht der Qualitätsbestimmung interessant ist.

Bei der Verwendung der DBLP und CiteSeer ist mit Einschränkungen in der Korrektheit und Vollständigkeit der Ergebnisse zu rechnen, deren Ursachen in der Datenqualität der Publikationsdienste liegen. Die Probleme der Datenqualität von Digitalen Bibliotheken resultieren vor allem daraus, dass oft unterschiedliche Schreibweisen des Namen einer Person in Gebrauch sind und Namen zudem nicht eindeutig sind, weil sie mehrere unterschiedliche Personen bezeichnen können. So existieren in den Systemen häufig mehrere Einträge zu ein und derselben Person, da der Name in den zugehörigen Publikationen in unterschiedlichen Schreibweisen vorliegt und das System die Zusammengehörigkeit nicht erkennt. Es kann also vorkommen, dass nicht alle Publikationen einer Zielperson extrahiert werden, oder keine gefunden werden, weil die Zielperson in dem System unter einer anderen Schreibweise geführt wird. Des Weiteren kann die Namensambiguität dazu führen, dass die Publikationen zweier Personen mit dem gleichen Namen unter einem Eintrag zusammengefasst sind. Dementsprechend werden auch Publikationen extrahiert, die nicht von der Zielperson stammen. Ähnliche Probleme kann die Namensambiguität bei der Integration der Einträge aus den beiden Systemen bereiten: Es

ist möglich, dass die Publikationseinträge in den beiden Quellsystemen von zwei verschiedenen Personen mit dem gleichen Namen stammen.

Informationsextraktion und -integration der Publikationsdaten

Zur Extraktion der Publikationseinträge der jeweiligen Zielperson aus den Autorenseiten der DBLP und den Ergebnisseiten von CiteSeer werden manuell erstellte Wrapper verwendet. Diese arbeiten in zwei Schritten: Zuerst konstruieren sie die URL der Ergebnisseite für die Suche nach der Zielperson im jeweiligen System, stellen eine Verbindung zu der Seite her und lesen den Inhalt. Im zweiten Schritt wenden sie die a priori festgelegten Extraktionsregeln auf den eingelesenen Quelltext an, um die relevanten Publikationsdaten zu erhalten.

In Hinblick auf die Extraktion der Veröffentlichungseinträge werden die Seiten beider Quellsysteme als strukturierte Webseiten klassifiziert. Die Extraktionsregeln nutzen dementsprechend die Strukturierungs- und Layouteigenschaften der Webseiten, die sie aus den verwendeten HTML-Tags ablesen, um die Zielinformationen zu lokalisieren.

Aus den Autorenseiten des DBLP-Servers werden die einzelnen Veröffentlichungseinträge extrahiert. Diese sind ebenfalls stark strukturiert, so dass die einzelnen Bestandteile eines Publikationseintrags identifiziert werden können. So wird zum einen die Integration der extrahierten Veröffentlichungen mit denen aus CiteSeer erleichtert, und zum anderen ist dadurch eine formatierte Darstellung der Ergebnisse möglich, bei der z.B. einzelne Bestandteile der Veröffentlichungen besonders hervorgehoben werden können.

Aus den Ergebnisseiten von CiteSeer werden pro Veröffentlichung der Titel, das komplette Label des Links auf die Veröffentlichungsseite, die URL der Veröffentlichungsseite und gegebenenfalls die Anzahl der Zitationen extrahiert.

Zur Integration der Veröffentlichungsdaten aus den beiden Quellen werden die Titel der extrahierten Veröffentlichungseinträge miteinander verglichen, um festzustellen welche Einträge in beiden Systemen gefunden wurden und diese in der Liste der Veröffentlichungen zu kennzeichnen. Zu den Veröffentlichungen, die nur in CiteSeer enthalten sind, sind nur die aus der Ergebnisseite extrahierten Informationen bekannt, so dass nicht der vollständige Veröffentlichungseintrag angegeben werden kann, sondern nur der Link auf die Veröffentlichungsseite. Dieser hat als Label den Titel und einen Teil der

Autorennamen. Eine Alternative dazu wäre, die einzelnen Veröffentlichungsseiten dieser Einträge aufzurufen, um die vollständigen Publikationsdaten zu erhalten. Dies wird jedoch als zu aufwendig eingestuft. Daher wurde die oben beschriebene Kompromisslösung umgesetzt. Der Link ist aus der Anwendung heraus zu öffnen, so dass der Nutzer die vollständige Publikationsseite einsehen kann.

3.2.2 Persönliche Homepages

3.2.2.1 Definition und Beschreibung

Die persönliche Homepage der Zielperson wird als Quelle für die in 3.1.1 festgelegten Attribute - akademischer Titel, Email-Adresse, wissenschaftliche Einrichtung oder Firma, Foto, Publikationsliste, Lebenslauf und Projekte - verwendet. Dabei wird unter der persönlichen Homepage eines Wissenschaftlers eine Webseite verstanden, die von dem Wissenschaftler selbst oder in seinem Auftrag erstellt wurde, mit dem Hauptziel, seine Arbeit und sich selbst vorzustellen [vgl. Hoff, Mundhenk 2001, 4]. In den meisten Fällen ist die Homepage Teil der Website des Arbeitgebers, z.B. einer Universität oder eines Forschungsinstituts.

Es gibt keine festen Regeln, die den Inhalt und die Struktur der Homepage eines Wissenschaftlers beschreiben, aber auf Grund von Erfahrungen und Beobachtungen in der Stichprobe wird angenommen, dass die „Besitzer“ relativ ähnliche Ziele haben, nämlich sich selbst und ihre Arbeit vorzustellen, auf Projekte und Publikationen zu verweisen und Kontaktinformationen bereitzustellen. Daher wird auch davon ausgegangen, dass die Homepages die gesuchten Informationen enthalten, wenngleich natürlich nicht immer jedes der Informationsitems vorhanden ist. Über diese inhaltliche Ähnlichkeit hinaus wird davon ausgegangen, dass es gewisse Konventionen bezüglich der Strukturierung von Homepages und der Verwendung von Begriffen zur Benennung der verschiedenen Inhaltselemente gibt, die sich durchgesetzt haben [vgl. Hoff 2002, 97]. Auf Grund dieser Aspekte wird die Menge der persönlichen Homepages von Informationswissenschaftlern als relativ homogene Datenmenge angesehen.

Die Regelmäßigkeiten im Aufbau und den verwendeten Begriffen werden zur Informationsextraktion aus den Homepages verwendet. Aufbauend auf der Analyse der Homepages der in 3.1.3 vorgestellten Referenzmenge und dem vorhandenen Erfahrungswissen wurden Heuristiken erstellt, die die Basis für die manuell erstellten Extraktions-

regeln bilden. Diese Regeln wurden in einem iterativen Prozess erstellt. Bei diesem Vorgehen wird eine Menge von Regeln auf den Trainingsdaten getestet. Das Extraktionsergebnis wird daraufhin überprüft, an welchen Stellen die Regeln zu speziell oder zu allgemein gefasst sind. Die Regeln werden dementsprechend modifiziert und der Prozess wiederholt, bis das Extraktionsergebnis als zufrieden stellend eingestuft wird. Diese Vorgehensweise wird in der IE-Forschung als *Knowledge Engineering Approach* [Appelt, Israel, 1999, 8] bezeichnet und insbesondere dann angewandt, wenn wie im vorliegenden Fall kein genügend großer annotierter Korpus zur Verfügung steht, um die Extraktionsregeln maschinell zu lernen. Dieser alternative Ansatz wird als *Automatic Training Approach* bezeichnet [Appelt, Israel, 1999, 8].

In den nächsten beiden Abschnitten werden die erstellten Heuristiken im Detail vorgestellt und ihre Anwendung in den Extraktionsregeln beschrieben.

3.2.2.2 Aufbau und Struktur

Die Hauptseite der Homepage eines Wissenschaftlers beinhaltet in der Regel den vollen Namen mit akademischem Titel, die Kontaktdaten inklusive E-Mail-Adresse und ein Foto des Wissenschaftlers. Diese Eigenschaft trifft auf 100% der Homepages der Referenzmenge zu. Weitere Angaben wie Informationen zur Lehre, den Mitarbeitern, Forschungsschwerpunkten und eben auch zu Publikationen und Projekten und der Lebenslauf werden, falls sie vorhanden sind, in der Regel auf zwei Arten dargestellt: Entweder sind sie auf von der Einstiegsseite verlinkten Unterseiten oder auf der Hauptseite in beliebiger Reihenfolge aufeinander folgend aufgeführt. In letzterem Fall werden in der Regel Zwischenüberschriften zur Strukturierung der Inhalte verwendet. Diese beiden Arten des Seitenaufbaus schließen einander nicht aus, es kommt durchaus häufig vor, dass einige der Daten auf der Hauptseite aufgeführt sind und andere Daten auf verlinkten Unterseiten, also Mischformen verwendet werden. Die erste Variante tritt in der Regel in Kombination mit einer Navigationsleiste auf, die die entsprechenden Links auf die Unterseiten enthält. In der Referenzmenge sind 50% der Seiten über Links auf Unterseiten strukturiert, 21% der Seiten sind sequentiell aufgebaut und 29% der Seiten sind Mischformen der beiden Ansätze.

Bezüglich der Seiteninhalte lassen sich Regelmäßigkeiten in der Bezeichnung von Inhaltselementen feststellen. So wird zur Benennung der Verweise auf Unterseiten und in den Zwischenüberschriften, mit denen die Seite bei einem sequenziellen Aufbau struk-

turiert wird, oft das gleiche Vokabular verwendet, d. h. die Menge der verwendeten Begriffe ist begrenzt und kann vorab ermittelt werden. Dazu werden die in den Seiten der Referenzmenge verwendeten Begriffe zusammengefasst und mit weiteren ergänzt, die erfahrungsgemäß ebenfalls verwendet werden. Inwieweit diese relativ umfassende, doch sicherlich keinesfalls vollständige Menge ausreichend ist, ist ein Untersuchungsgegenstand der Evaluierung (siehe Kapitel 4).

Wie dieses Wissen in der Extraktion der einzelnen Informationsitems eingesetzt wird, wird im nächsten Kapitel für die einzelnen zu extrahierenden Informationsobjekte beschrieben.

3.2.2.3 Informationsextraktion aus der Homepage

Akademischer Titel

Zur Extraktion des akademischen Titels der Zielperson wurde eine Liste mit im Deutschen verwendeten akademischen Titeln erstellt. Diese Liste enthält die folgenden Einträge: *Prof. Dr., Prof., Dr., MA, M.A., Dipl.-Inf., Dipl.-Ing., Dipl. Inf., Dipl. Ing., Ph.D.*. Es wurde versucht, alle bekannten Schreibweisen akademischer Titel zu berücksichtigen. Um den konkreten Titel einer Zielperson zu finden, werden die Wörter der Hauptseite mit den in der Liste enthaltenen Titelabkürzungen verglichen. Da angenommen wird, dass der akademische Titel im Titel der Homepage oder am Anfang der Hauptseite aufgeführt wird, wird das erste Vorkommen eines akademischen Titels in einer Seite als der Titel der Zielperson gewertet.

E-Mail-Adresse

Das Verfahren zur Extraktion der E-Mail-Adresse der Zielperson beruht auf der Annahme, dass der lokale Teil[18] der E-Mail-Adresse eines Wissenschaftlers den Nachnamen oder einen Teil des Nachnamens beinhaltet. So werden die in der Hauptseite enthaltenen E-Mail-Verweise auf eben diese Eigenschaft überprüft. Wird eine E-Mail-Adresse gefunden, die die Anforderung erfüllt, wird davon ausgegangen, dass es sich um die E-Mail-Adresse der Zielperson handelt.

[18] eine E-Mail-Adresse setzt sich aus einem lokalen Teil und einem Domain-Teil zusammen, die durch das @-Zeichen getrennt sind.

Foto der Zielperson

Zur Identifikation des Fotos der Zielperson auf der Hauptseite wurden Regelmäßigkeiten in der Benennung der Bilddateien und der verwendeten Alternativtexte untersucht: Häufig wir der komplette Name oder mindestens der Nachname als Dateiname verwendet (dies ist in 64% der Seiten der Referenzmenge der Fall). Der Alternativtext wird weniger häufig verwendetet (in 50% der Seiten der Referenzmenge) und enthält entweder den Namen bzw. Nachnamen oder Schlüsselwörter wie „Foto“ oder „Bild“. Diese Regelmäßigkeiten werden zur Identifikation des Fotos der Person genutzt, indem zunächst die Dateinamen der in der Homepage vorhandenen Bilder auf Vorkommen des Nachnamens geprüft werden. Ist dies nicht erfolgreich, werden die Alternativtexte der Grafiken auf die Verwendung des Nachnamens oder eines der Schlüsselwörter hin überprüft. So kann die URL des Speicherortes des Fotos identifiziert werden.

Institution/Firma

In der Festlegung des Suchziels wurde über die bisher beschriebenen Informationsitems hinaus, die Institution oder Firma, für die die Zielperson tätig ist, als interessantes Informationsobjekt festgelegt. Doch diese Information kann nicht mit einfachen Regeln ermittelt werden. Denn in den meisten Fällen ist der Name der Einrichtung, für die die Zielperson tätig ist, nicht explizit aufgeführt, sondern implizit in einem Logo auf der Homepage enthalten. Um den Namen aus solchen Grafiken zu extrahieren müssten Verfahren der *Optical Character Recognition* (OCR) angewandt werden. In seltenen Fällen ist der Name der Einrichtung explizit angegeben, wenn eine Postanschrift aufgeführt ist. In solchen Fällen bedarf es komplexerer Regeln zur Eigennamenerkennung. Die Erstellung eines solchen Verfahrens bzw. die Einbindung eines OCR-Verfahrens kann im Rahmen dieser Arbeit nicht geleistet werden.

Akademischer Titel, E-Mail-Adresse und das Foto können als singuläre Informationen bezeichnet werden. Sie sind mit Hilfe von Regulären Ausdrücken und Stringvergleichen zu extrahieren. Die Informationen sind nicht in weitere Informationsitems zerlegbar und somit sind Anfang und Ende des zu extrahierenden Informationsobjekts genau identifizierbar. Im Gegensatz dazu stehen die zusammenhängenden Informationen: Sie bestehen aus mehreren Zeilen Text und können in weitere Informationsitems zerlegt werden. Bei diesen ist die Bestimmung der genauen Grenzen des zu extrahierenden Bereichs problematisch. Dazu gehören die Publikations- und Projektlisten und der Lebenslauf.

Publikationen, Projekte, Lebenslauf

Wie in 3.2.2.2 beschrieben, sind die Angaben zu Publikationen und Projekten und der Lebenslauf, falls vorhanden, entweder auf der Hauptseite der Homepage aufgeführt oder auf von der Hauptseite aus verlinkten Unterseiten. Da das System nicht in der Lage ist, die Art der Strukturierung der Homepage zu erkennen und zudem häufig Mischformen verwendet werden, wird für jede der Zielinformationen geprüft, ob und in welcher Art und Weise sie dargestellt ist. Dazu wird eine Liste verwendet, welche die typischerweise zur Bezeichnung der Inhaltselemente verwendeten Schlüsselwörter enthält: Für die Angaben zu Veröffentlichungen sind das die Begriffe „Publikationen“ und „Veröffentlichungen“, für die Angaben zu Projekten „Projekte“ und „Forschungsprojekte“ und für den Lebenslauf die Bezeichnungen „Lebenslauf“, „Vita“ und „Zur Person“.

Die Hauptseite wird nach diesen Begriffen durchsucht. Wird ein Begriff gefunden, so wird festgestellt, ob es sich um das Label eines Verweises handelt oder um ein normales Textelement. Handelt es sich um ein Label, so wird der zugehörige Link extrahiert und die entsprechende Unterseite aufgerufen. Bezüglich des Aufbaus einer Unterseite wird davon ausgegangen, dass der Begriff als Überschrift in dieser wieder verwendet wird und die gesuchte Information auf diese Überschrift folgt. Als Zielinformation wird dementsprechend der auf die Überschrift folgende Text extrahiert.

Ist eine Seite vollständig bzw. zum Teil sequenziell aufgebaut, wird davon ausgegangen, dass sie über einheitlich formatierte Unterüberschriften strukturiert ist. Das Ende eines zu extrahierenden Bereichs ist dann entweder durch die Überschrift des Folgebereichs definiert oder durch das Seitenende, wenn es sich um den letzten Eintrag auf der Seite handelt. Werden also eine oder mehrere relevante Zwischenüberschriften gefunden, können alle folgenden Zwischenüberschriften der Annahme nach über die Formatierung bestimmt werden. Der zu extrahierende Zieltext wird also von einer Zwischenüberschrift, die eines der relevanten Schlüsselwörter enthält, eingeleitet und durch die darauf folgende Zwischenüberschrift begrenzt.

Ungelöst ist bei diesem Vorgehen das Problem der eindeutigen Bestimmung des Endes der Bereiche, die extrahiert werden sollen. Ist die Information auf einer Unterseite enthalten oder der letzte Eintrag einer sequenziell aufgebauten Seite, so kann nicht bestimmt werden, wo das Ende der relevanten Daten ist und unter Umständen wird so z.B. die Fußzeile der Seite mitextrahiert.

3.2.3 Spezialisierte Suche nach Homepages

Für die automatisierte Suche nach der Homepage der jeweiligen Zielperson wurde ein eigenes Suchverfahren entwickelt, das auf die Suche nach Homepages von Informationswissenschaftlern spezialisiert ist.

Suchmaschinen mit ähnlichem Ziel sind bereits entwickelt und auch im Internet öffentlich zugänglich gemacht worden. Die zwei bekannten Systeme sind *HPSearch*[19], eine Suchmaschine für persönliche Homepages von Informatikern und *Ahoy! The Homepage Finder*[20], ein System zur Suche nach persönlichen Homepages, ohne Einschränkung auf einen bestimmten Kreis von Zielpersonen. Jedoch entfällt die Möglichkeit, diese Suchmaschinen über entsprechende Wrapper in den Prototypen einzubinden, da der Betrieb von Ahoy! aus Personalmangel eingestellt wurde und der Index von HPSearch aus demselben Grund nicht mehr aktualisiert wird.

Darüber hinaus gab es im Rahmen der *TREC*-Initiative[21] 2001 einen Web Track, der sich mit dem „Homepage Finding Task“ befasst hat. Unter Homepage wird in der Aufgabe aber eine „site entry page“ [Hawking, Craswell 2002, 61] verstanden, und nicht ausschließlich die persönliche Homepage einer Person, wie in den im nächsten Abschnitt beschriebenen Suchverfahren und der vorliegenden Arbeit. Aufgrund dieser anderen Aufgabenstellung werden die im Rahmen des Web Track evaluierten Suchansätze nicht näher betrachtet. Ein Ergebnis der Evaluierung ist aber auch für die ausschließliche Suche nach Homepages von Bedeutung: Es hat sich herausgestellt, dass Verfahren, die den URL Text in die Retrieval-Methode einbeziehen, wesentlich bessere Ergebnisse erzielen, als solche, die rein inhaltbasierte Methoden verwenden [Hawking, Craswell 2002, 64].

Im Folgenden werden die Vorgehensweisen von Ahoy! und HPSearch beschrieben und das implementierte Suchverfahren, das auf den Ergebnissen dieser Forschungsarbeiten aufbaut, vorgestellt.

[19] http://hpsearch.uni-trier.de/ (verifiziert am 29.03.2006)
[20] http://www.cs.washington.edu/research/projects/WebWare1/www/ahoy/about/index.htm (verifiziert am 29.03.2006)
[21] Mit der Text Retrieval Conference (TREC) stellt das National Institute of Standards and Technology (NIST) eine Plattform zur einheitlichen Evaluierung von IR-Systemen zur Verfügung. Mittlerweile werden neben den IR-Systemen im klassischen Sinne auch andere IR-Anwendungen evaluiert. Diese verschiedenen Aufgaben werden als Tracks bezeichnet.

3.2.3.1 Ansätze für Spezielle Suchmaschinen für Homepages

Ahoy! The Homepage Finder

Ahoy! The Homepage Finder [Shakes et al. 1997] ist eine Suchmaschine für persönliche Homepages, wobei keinerlei Einschränkung bezüglich der Gruppe der Zielpersonen vorgenommen wird. Das angewandte Verfahren wird als „*Dynamic Reference Shifting*" bezeichnet [Shakes et al. 1997, 1]: Aus einer Menge von potentiellen Homepages wird mit Hilfe von Heuristiken und Informationen aus externen Datenbanken zu E-Mails und URLs versucht, die Homepage der Zielperson herauszufiltern. Ist dieses Vorgehen nicht erfolgreich, so wird, falls der Nutzer zusätzlich zu dem Namen der Zielperson noch eine Institution angegeben hat, versucht, die URL der Homepage zu erraten. Dazu wird in früheren, erfolgreichen Suchen gesammeltes Wissen über die Bildung von Homepage-URLs an verschiedenen Institutionen verwendet.

Die genaue Arbeitsweise ist wie folgt: Der Nutzer stellt seine Anfrage in Form von Vor- und Nachnamen der Zielperson und hat zusätzlich die Möglichkeit, eine Institution und ein Land anzugeben. Der Name wird als Anfrage an die Metasuchmaschine *MetaCrawler*[22] gestellt. Parallel dazu werden Anfragen an E-Mail-Verzeichnisdienste gerichtet und in einer internen Datenbank nach der URL der Institution, falls sie angegeben wurde, gesucht. Mit Hilfe dieser orthogonalen Informationen werden irrelevante Ergebnisse aus der Liste der von MetaCrawler zurückgelieferten Suchergebnisse entfernt. Darüber hinaus werden mit einem heuristischen Verfahren Titel, URL und Snippet[23] der einzelnen Ergebniselemente untersucht. Die verwendeten Heuristiken beschreiben typische Eigenschaften von Homepages, werden aber in der Systembeschreibung nicht näher spezifiziert. Das Ergebnis dieser Evaluierung ist eine nach Relevanz geordnete Liste der verbliebenen Suchergebnisse, die dem Nutzer präsentiert wird.

Wenn eine Suche erfolgreich ist, dann extrahiert das System die Regeln zur Generierung der Homepages der angegebenen Institution. Es lernt, wie die Basis URL der Institution ist und wie jeweils der personenbezogene Teil der URL gebildet wird. So wird eine Wissensbasis erstellt und mit jeder erfolgreichen Suche erweitert. Führt das Filtern der Ergebnisse zu keinem Treffer und hat der Nutzer eine Institution angegeben, so wird versucht mit den Informationen der Wissensbasis die URL zu generieren.

[22] http://www.metacrawler.com (verifiziert am 29.03.2006)

[23] Mit dem Begriff *Snippet* wird im Englischen ursprünglich ein Schnipsel bezeichnet. Damit wird aber auch der Vorschautext in der Google-Ergebnisseite bezeichnet, der die an den Suchbegriff angrenzenden Wörter aus der jeweiligen Seite enthält.

HPSearch

HPSearch ist eine spezialisierte Suchmaschine für persönliche Homepages von Informatikern. Das von Gerd Hoff im Rahmen seiner Doktorarbeit an der Universität Trier [Hoff 2002] entwickelte Verfahren umfasst folgende Schritte: Zu einem gegebenen Namen wird mit Hilfe von „normalen“ Suchmaschinen eine Menge potentieller Homepages gesammelt und in einem zweistufigen Verfahren evaluiert. Zuerst wird jede der Seiten auf Basis der von den Suchmaschinen gelieferten Informationen bewertet. Es werden also die URL, der Titel der Seite, das Snippet und die Position im Ranking der entsprechenden Suchmaschine berücksichtigt. Aufbauend auf dieser Bewertung wird ein Ranking erstellt und die am höchsten gerankten Seiten werden in der zweiten Stufe des Verfahrens aufgerufen, damit eine detailliertere Bewertung durchgeführt werden kann. Zusätzlich zu den im ersten Schritt evaluierten Bestandteilen werden der komplette Text der Homepage unter Berücksichtigung der verschiedenen Tags und einige numerische Eigenschaften wie die Größe der Homepage untersucht. Das finale Ranking beruht auf den Ergebnissen dieser zweiten Bewertung und wird zusammen mit dem Suchnamen in einer Datenbank gespeichert. Der Nutzer kann über ein Webinterface nach der Homepage einer Person mit dem Namen als Suchterm suchen und erhält als Ergebnis die vorab ermittelten und gerankten Ergebnisse in einem Hypertext Dokument.

Bei HPSearch handelt es sich also um eine indexbasierte Lösung: Die Homepages werden nicht wie bei Ahoy! und dem erstellten Prototyp erst zur Anfragezeit sondern vorab ermittelt.

Zur Erstellung der Bewertungsfunktion werden signifikante Eigenschaften von Homepages von Informatikern ermittelt. Dazu werden in einer Menge von Homepages relevante Eigenschaften identifiziert und die Signifikanz dieser Eigenschaften über einen Vergleich mit einer Kontrollmenge beliebiger Webseiten festgestellt [vgl. Hoff 2002, 93]. Die aus dieser Analyse resultierende Bewertungsfunktion umfasst ca. 500 boolesche und numerische Eigenschaften [vgl. Hoff, Mundhenk 2001, 5]. Die Anzahl der Eigenschaften ist deswegen so hoch, da auch das Vorkommen eines bestimmten Begriffs, wie z.B. „Informatik“, als eine einzelne Eigenschaft gewertet wird.

Die booleschen Eigenschaften beziehen sich vor allem auf das Vorkommen bestimmter Wörter, Abkürzungen und des Namens der Zielperson im Bereich verschiedener HTML-Tags der Seite und in der URL. Bei der Analyse der Seite selbst wird zwischen Title-Tag, Überschriften-Tags, Label-Tags und dem sonstigen Text unterschieden. Signi-

fikante Eigenschaften sind dabei u.a. das Vorkommen von „home“ oder „homepage“ und die Nennung des Namens (Vor- und Nachname, oder nur Nachname) im Titel oder der ersten Überschrift einer Seite. Aus der Analyse der URL resultieren folgende signifikante Eigenschaften: Zum einen das Vorkommen eines Instituts- oder Fachbereichskürzel wie „cs“, „db“ oder „uni“ und das Auftreten von Dateibezeichnungen wie „home“ oder „index“. Zum anderen die Verwendung des Namens bzw. Teilen des Namens der Zielperson, häufig in Kombination mit dem Tilde-Zeichen oder Bezeichnern wie „people“ oder „staff“.

Zur Bestimmung numerischer Eigenschaften wurden unter anderem die Größe der Seiten und die Länge der URLs untersucht. Besonders auffallend ist dabei die verhältnismäßig geringe Größe der Homepages. Über diese Eigenschaften, die in der Bewertungsfunktion zum Tragen kommen, hinaus wurden im Laufe der Evaluierung u.a. Dokumente entfernt, die den Nachnamen der Zielperson nicht enthalten, und solche Dokumente, „die von bestimmten Web-Diensten, welche Homepage-ähnliche Dokumente verwalten (z.B. DBLP […]), stammen“ (Hoff 2002, 123].

Von Hoff wurde zudem untersucht, wie sich die ermittelten Eigenschaften im Laufe der Zeit verändert haben [vgl. Hoff 2002, 151ff.]. Dazu wurde eine vergleichende Untersuchung für die Jahre 1998, 1999, 2000 und 2001 gemacht. Dabei konnten zwar einige signifikante Änderungen festgestellt werden, insbesondere in der Menge der in den Homepages verwendeten Wörtern, aber insgesamt beobachtete Hoff eine „[…] relativ große Konstanz bei den Eigenschaften der persönlichen Homepages von Informatikern“ [Hoff 2002, 163].

Da nicht alle von Hoff bewerteten Eigenschaften im Rahmen dieser Beschreibung vorgestellt werden können, wurde lediglich ein Überblick über die durchgeführten Analysen und eine Zusammenfassung der wichtigsten Ergebnisse gegeben. Dabei lag der Schwerpunkt auf den Eigenschaften, die in den Zusammenfassungen der Suchmaschinen ermittelt werden. Diese sind für die Entwicklung des Prototypen relevant, da dieser nicht die ganzen Seiten, sondern nur die von der Suchmaschine bereitgestellten Zusammenfassungen bewertet, wie im nächsten Kapitel beschrieben wird.

3.2.3.2 Realisiertes Verfahren

Die in Ahoy! und HPSearch implementierten Suchverfahren haben sich als erfolgreich erwiesen. In einer Evaluierungsstudie [vgl. Hoff, Mundhenk 2001, 11] fand HPSearch in 84% der Fälle die richtige Homepage. Im Vergleich dazu führte in der gleichen Untersuchung die beste Suchmaschine lediglich in 60% der Anfragen die Homepage an erster Stelle auf. In einem ähnlichen Vergleich der Suchergebnisse von Ahoy! mit denen herkömmlicher Suchmaschinen lieferte Ahoy! in 74% der Fälle die Homepage der Zielperson an erster Stelle und erzielte damit mit Abstand die höchste Precision. AltaVista hingegen fand als beste Suchmaschine 58% der Homepages und nannte 23% von diesen an erster Stelle (vgl. Shakes et al. 1997, 1].

Beide Verfahren nutzen existente Suchmaschinen, um eine Vorauswahl potenzieller Homepages aus der Menge aller Webseiten zu erhalten. Um aus diesen Suchmaschinenergebnissen die gesuchte Homepage herauszufiltern, verwenden beide Systeme Heuristiken, die charakteristische Eigenschaften von Homepages beschreiben, sowie ein auf diesen Heuristiken basierendes Rankingverfahren. Aufbauend auf diesen Ergebnissen wird in dem entwickelten Prototypen ein ähnliches Verfahren zur Umsetzung der spezialisierten Suche nach Homepages eingesetzt. Es werden charakteristische Eigenschaften von Homepages von Informationswissenschaftlern ermittelt und in eine Bewertungsfunktion überführt, die dazu verwendet wird, aus den über die Suche mit einer Suchmaschine ermittelten Homepage-Kandidaten die Homepage der Zielperson herauszufiltern.

Ahoy! verwendet, wie weiter oben beschrieben, zudem Informationen aus externen Quellen, um Ergebnisse, von denen vermutet wird, dass sie nicht die gesuchte Homepage sind, vorab auszuschließen. Zu diesen Quellen gehören u.a. E-Mail-Verzeichnisdienste. Für den deutschsprachigen Raum sind derartige Quellen nicht bekannt, so dass diese Methode nicht auf das entwickelte System übertragbar ist.

Bevor die Entscheidung für die letztendlich umgesetzte Lösung getroffen wurde, ist zunächst versucht worden, die Position der Homepages im Ranking der Suchmaschinenergebnisse durch Erweiterung der Anfrage bestehend aus Vor- und Nachname der Zielperson zu verbessern. Die Erweiterungsterme wurden intuitiv gewählt und waren z.B. „Homepage“ und „Informationswissenschaft“. Dies führte jedoch nicht durchgängig zu besseren Ergebnissen und auch dazu, dass bei der Suche mit der unerweiterten Anfrage an erster Stelle aufgeführte Treffer bei der erweiterten Suche schlechter gerankt

wurden. Daher wurde von dieser Lösung abgesehen. Hoff hat im Zuge der Entwicklung von HPSearch die gleiche Beobachtung gemacht und sich daher auch gegen eine Anfrageerweiterung entschieden [vgl. Hoff 2002, 131]. Im Ausblick wird jedoch eine Möglichkeit der Anfrageerweiterung skizziert, die nicht auf vorab festgelegten Erweiterungstermen beruht.

Als Suchmaschine wird *Google*[24] verwendet. Für Google sprechen zwei Aspekte: Es wird angenommen, dass Google mit ca. 8 Milliarden indexierten Seiten über den größten Index der bekannten Suchmaschinen verfügt [vgl. Lewandowski 2005b, 42]. Darüber hinaus stellt Google eine *API*[25] zur Verfügung, um externen Applikationen die Einbindung seiner Dienste zu ermöglichen. Somit entfällt der Aufwand für die Generierung eines Wrappers für die Google-Ergebnisseite und die Verwendung der Google Ergebnisse ist unabhängig von Formatänderungen in den Google Seiten.

Zur Bewertung der Ergebnisse werden nur die von Google zu einem Ergebnis gelieferten Daten berücksichtigt. Diese sind der Titel der Seite, die URL und das von Google erstellte Snippet. Es werden also keine kompletten Webseiten in die Bewertung miteinbezogen. Ein praktischer Grund für diese Entscheidung ist, dass eine enorme Zeitersparnis erreicht wird, da die Anzahl der Seitenaufrufe zur Suchzeit möglichst gering gehalten und der Analyseaufwand ebenfalls minimiert wird. Aber auch die Ergebnisse anderer Untersuchungen sprechen dafür, dass eine Analyse der kompletten Seiten nicht zu besseren Ergebnissen führt. Das erfolgreiche Verfahren von Ahoy! berücksichtigt lediglich die von den Suchmaschinen zu den jeweiligen Seiten gelieferten Informationen. Xi et al. [2002], die ein Verfahren zur Umsetzung des Homepage Finding Task im Sinne der Definition des TREC 10 Web Tracks entwickelt haben, kommen zu dem Schluss, dass die Retrievalverfahren wesentlich bessere Ergebnisse erzielen, wenn sie Titel, Ankertexte und ein Abstract der Webseiten verwenden, als wenn sie den Volltext einer Seite nutzen. Auch Hoff [2002, 131] stellt fest, dass die Einbeziehung der Volltexte der Seiten in die Bewertung, also die zweite Stufe des Bewertungsverfahren, in den seltensten Fällen das Ergebnis der Vorsortierung ändert.

[24] http://www.google.com (verifiziert am 29.03.2006)

[25] API steht für *Application Programming Interfaces* und bezeichnet die von einem System zur Nutzung durch andere Systeme zur Verfügung gestellte Schnittstelle. Anders ausgedrückt, "the term API refers to a set of functions that a developer can call on to perform application tasks. For example, opening a file requires use of one or more functions provided by the operating system API." [Mueller 2004, 4]

Vorauswahl durch Google

Als Anfrageterm für die Suche mit Google werden Vor- und Nachname der Zielperson benutzt. Diese werden über die Verwendung von Anführungszeichen an Anfang und Ende gemäß der Google-Syntax als Phrase gekennzeichnet. So wird erreicht, dass nur nach Dokumenten gesucht wird, die den vollständigen Namen der Zielperson enthalten. Dem liegt die Annahme zu Grunde, dass eine persönliche Homepage den vollständigen Namen des Besitzers enthält. Diese Annahme trifft auf alle Seiten der Referenzmenge zu.

Google bietet außerdem die Möglichkeit, bestimmte Dokumenttypen von der Suche auszuschließen. Da bei der Suche mit den Namen von Wissenschaftlern häufig Veröffentlichungen und Präsentationen dieser Wissenschaftler in Form von PDF-, PostScript- und PowerPoint-Dokumenten unter den zurückgelieferten Ergebnissen sind, werden diese über eine entsprechende Erweiterung des Anfrageterms von der Suche ausgeschlossen. Darüber hinaus wird gemäß der Spezifikation der Suchaufgabe von vornherein festgelegt, dass nur nach deutschsprachigen Dokumenten gesucht werden soll. Die Suchfunktion bietet die Möglichkeit der Einschränkung der Suche auf einen oder mehrere Sprachräume.

Die Suche mit Google als WebService liefert äquivalent zur normalen Suche im Web eine gerankte Liste von zehn Webseitenreferenzen. Um mehr als die ersten zehn Ergebnisse zu erhalten, muss eine neue Suche aufgerufen werden, wobei angegeben werden muss, ab welchem Ergebnis man die nächsten zehn Ergebnisse erhalten möchte. Für die Suche nach der Homepage werden nur die ersten zehn Ergebnisse berücksichtigt. Es ist folglich nur ein Aufruf des Google WebService notwendig. Zum einen spart dies Zeit, zum anderen hat sich in der Analyse der Referenzmenge gezeigt, dass 13 aus 14 Seiten unter den ersten zehn Google-Ergebnissen sind. In dem Fall, in dem die Homepage nicht unter den ersten zehn Google-Ergebnissen ist, ist der Name der Zielperson stark überlagert, da es neben dem Informationswissenschaftler einen gleichnamigen deutschen Philosophen und einen gleichnamigen deutschen Schauspieler gibt.

Bewertungsverfahren

Die Vorgehensweise bei der Entwicklung der Bewertungsmethode orientiert sich am Vorgehen und den Ergebnissen von Hoff [2002]. URLs, Titel und Snippets der in der

Referenzmenge enthaltenen Homepages werden auf das Vorkommen des Namens der Zielperson, bestimmter Zeichenketten und Begriffen aus der Informationswissenschaft untersucht. Dabei werden von Hoff identifizierte Eigenschaften in der Referenzmenge überprüft. Sie werden zusammengefasst oder vereinfacht, bzw. an die Informationswissenschafts-Domäne entsprechend der Eigenschaften der Seiten der Referenzmenge angepasst. Zusätzlich werden weitere Eigenschaften ermittelt und im Laufe der ständigen Evaluierungen gemachte Beobachtungen in das Bewertungsverfahren miteinbezogen. Im Folgenden werden die einzelnen Eigenschaften aufgelistet. Titel und Snippet werden bei der Bewertung zusammengefasst betrachtet[26].

Eigenschaften der URL

- Vorkommen eines Institutionskürzels: *uni*, *fh*, *tu*
- Vorkommen eines der Schlagwörter *people*, *staff*, *person* oder *name*
- Vorkommen eines Fachbereichskürzels: *iw*, *inf-wiss*, *info*, *informatik*, *hci*
- Vorkommen einer Tilde (~)
- Vorkommen des Nachnamens der Zielperson oder von Teilen des Nachnamens (es werden Trigramme untersucht)

Eigenschaften von Snippet und Titel zusammen

- Vorkommen von *home*, *homepage*, *home page*
- Vorkommen von Fachbereichsbezeichnungen: *Informatik*, *Informationswissenschaft*, *IW*
- Vorkommen eines akademischen Titels: *Prof. Dr.*, *Prof.*, *Dr.*, *MA*, *M.A.*, *Dipl.-Inf.*, *Dipl.-Ing.*, *Dipl. Inf.*, *Dipl. Ing.*, *Ph.D.*

Als weitere Eigenschaft wird das Vorkommen des Nachnamen der Zielperson im Titel gewertet.

[26] Sind Nomen klein geschrieben, so ist das beabsichtig, da es sich z.B. um Zeichenketten in der URL handelt. Bei den Eigenschaften von Titel und URL ist die Überprüfung der Eigenschaften nicht abhängig von Groß- und Kleinschreibung.

Berechnung der Bewertung

Alle aufgezählten Eigenschaften werden als boolesche Eigenschaften gewertet. Die Bewertungsfunktion summiert für jedes Ergebnis die Werte für die einzelnen Eigenschaften auf, wobei die verschiedenen Eigenschaften nicht gewichtet werden. Ob und wie eine Gewichtung der einzelnen Eigenschaften sinnvoll wäre, ist in einer detaillierteren Evaluierungsstudie zu untersuchen. Zusätzlich zu den genannten Eigenschaften wird auch die Position des einzelnen Ergebnisses im Ranking von Google mit in die Bewertung einbezogen. Dazu wird jeder Position ein Wert zwischen 1 und 0 zugewiesen. Der beste Ergebnis bekommt den Wert 1, das zweitbeste Ergebnis den Wert 0,9 usw..

Bevor die einzelnen Suchergebnisse evaluiert werden, werden bestimmte Elemente, von denen angenommen wird, dass sie garantiert oder mit großer Wahrscheinlichkeit nicht die gesuchte Homepage sind, aus der Ergebnismenge entfernt. Dazu gehören:

- Seiten, bei denen der Nachname oder ein Teil des Nachnamens weder in der URL noch im Titel vorkommen.
- Seiten, die von Digitalen Bibliotheken und ähnlichen Diensten stammen. Dazu gehören CiteSeer, DBLP und ACM Digital Library. Diese Elemente werden über einen Vergleich der URLs identifiziert.

Eliminierung veralteter Seiten

Während der fortlaufenden Evaluierung hat sich herausgestellt, dass häufig veraltete Homepages der Zielperson, z.B. von früheren Arbeitgebern, weiterhin online sind und dementsprechend in den Suchergebnissen enthalten sind. Diese Seiten werden gemäß dem Ziel der Bewertungsfunktion natürlich hoch bewertet und können der aktuellen Homepage „Konkurrenz machen“. Daher wurde ein Verfahren entwickelt, um veraltete Seiten von dem Ranking auszuschließen.

Auf das Aktualisierungsdatum der einzelnen Seiten kann nicht zurückgegriffen werden, da die einzelnen Seiten aus Gründen der Effizienz nicht aufgerufen werden. Zudem wurde bei der Evaluierung dieser Möglichkeit festgestellt, dass nicht auf allen Seiten das Aktualisierungsdatum angegeben ist und selbst wenn vorhanden häufig nicht aktuell ist. Aus diesem Grund wurde eine Methode entwickelt, die die Rangfolge akademischer Titel verwendet, um veraltete Seiten zu identifizieren: Es wird angenommen, dass der höchste akademische Grad, der in den Ergebnisseiten einer Suche gefunden wurde, der

aktuelle akademische Titel der Suchperson ist. Alle Seiten, die einen niedrigeren akademischen Titel enthalten, werden als nicht aktuell angesehen und im Ranking nicht berücksichtigt. Seiten, in denen kein akademischer Titel vorkommt, sind von dieser Prozedur ausgeschlossen und werden in Ranking einbezogen. In Fällen, in denen der akademische Titel im Laufe der Zeit unverändert geblieben ist, greift diese Methode nicht, so dass eine veraltete Seite, die den aktuellen Titel der Zielperson enthält, nicht identifiziert werden kann.

Ranking der Ergebnisse

Nach der Berechnung der Bewertung für die einzelnen Ergebnisseiten und dem Ausschluss nicht aktueller Seiten wird ein Ranking auf der Basis der Bewertungen der einzelnen Seiten erstellt. Die Seite, die an erster Stelle gerankt ist, wird als gesuchte Homepage angesehen, wenn sie mit mindestens vier Punkten bewertet wurde. Dieser Schwellenwert wurde eingerichtet, um zu vermeiden, dass fälschlicherweise eine Seite, bei der es sich nicht um die Homepage der Zielperson handelt, als Homepage gewertet wird. Dieser Fall würde ansonsten eintreten, wenn die eigentliche Homepage der Zielperson nicht unter den Google-Ergebnissen ist. Der Schwellenwert wurde experimentell auf Basis der Referenzmenge festgelegt, in der alle Homepages mit mindestens vier Punkten bewertet worden sind. Natürlich kann dieser Schwellenwert nicht garantieren, dass nur Seiten als Homepage klassifiziert werden, die tatsächlich eine Homepage sind.

3.3 Prozessorientierter Systemüberblick

Abbildung 1 zeigt die einzelnen Komponenten des Systems und ihr Zusammenspiel sowie den Informationsfluss während eines Suchvorgangs. Im Folgenden soll dies zusammenfassend erläutert werden. Es wird die interne Abfolge der Suche für den Best Case beschrieben, d. h. für den Fall, dass in allen Quellen die gesuchten Informationen gefunden werden.

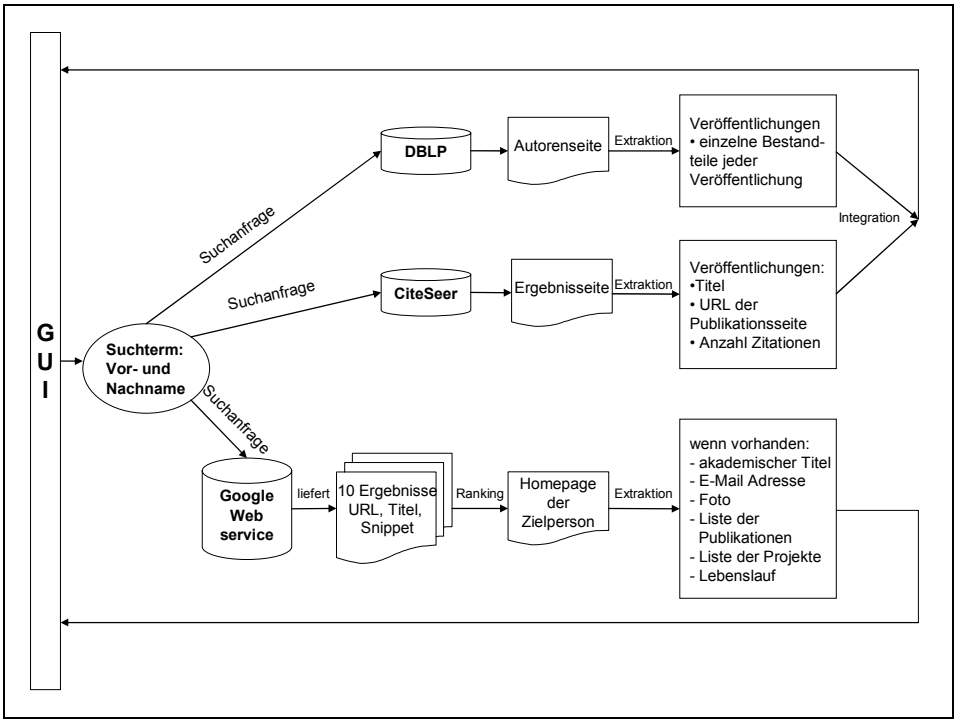

Abbildung 1: Systemüberblick

Der Benutzer gibt als Anfrage den Namen der Person ein, zu der er Informationen sucht. Mit diesem Namen wird die URL der Autorenseite der DBLP dieser Person generiert und die Seite aufgerufen. Die einzelnen Veröffentlichungseinträge werden extrahiert und in einem weiteren Schritt die einzelnen Bestandteile dieser Veröffentlichungen identifiziert.

Als nächstes wird die URL der Ergebnisseite für die Suche in CiteSeer mit dem Namen der Person als Anfrageterm erstellt und eine Verbindung zu der Seite hergestellt. Die Titel der aufgeführten Veröffentlichungen werden zusammen mit der URL der einzelnen Veröffentlichungsseite und, falls angegeben, der Anzahl der Zitationen extrahiert. Die in den beiden Quellen gefundenen Publikationen werden zu einer Liste zusammengefügt. Dabei werden in beiden Systemen aufgeführte Publikationen identifiziert und gekennzeichnet.

Der nächste Schritt ist das Auffinden der Homepage der Zielperson wozu eine Google-Suche mit dem Namen der Zielperson als Anfrageterm durchgeführt wird. Das zurückgelieferte Suchergebnis beinhaltet eine gerankte Liste von zehn Webseitenreferenzen. Diese Referenzen bestehen jeweils aus der URL, dem Titel der referenzierten Seite und dem von Google erstellten Snippet. Auf Basis dieser Daten werden die potentiellen Homepages hinsichtlich des Vorhandenseins typischer Eigenschaften bewertet und ein Ranking erstellt. Die am höchsten gerankte Seite wird als Homepage der Zielperson gewertet und aufgerufen.

Aus der Homepage werden die gesuchten Informationen zu der Zielperson, also akademischer Titel, E-Mail-Adresse, Foto, Liste der Publikationen, Liste der Projekte sowie der Lebenslauf, falls diese Angaben gemacht werden, extrahiert.

Im letzten Schritt werden die in der Homepage gefundenen Informationen zusammen mit den Publikationsdaten aus der DBLP und CiteSeer dem Nutzer in strukturierter Form in der Benutzeroberfläche präsentiert. Dazu werden einzelne HMTL-Seiten, die der Nutzer über ein Karteireitersystem auswählen kann, generiert (siehe Kapitel 4.1).

Die Abfolge der Suchen in den einzelnen Quellsystemen und die darauf folgende Informationsextraktion ist zufällig gewählt worden und bis auf die nicht aufzuhebende Reihenfolge der Suche nach der Homepage und der Informationsextraktion aus dieser nicht zwingend einzuhalten. Eine Parallelisierung der Teilaufgaben ist auch möglich und für eine Verringerung der Suchzeit sinnvoll.

4 Das implementierte System

In diesem Kapitel wird die Implementierung der entwickelten Verfahren beschrieben. Der Prototyp ist in Form einer Java-Applikation umgesetzt worden. Dazu wurde Java v.1.4.2.02 verwendet. Als Entwicklungsumgebung wurde Eclipse[27] in der Version 3.1.1. genutzt.

Zunächst wird die Benutzeroberfläche vorgestellt und ihre Funktionalität erläutert. Im zweiten Teil werden der Aufbau und die Funktionsweise des erstellten Java-Programms behandelt.

4.1 Grafische Benutzeroberfläche

Die Benutzeroberfläche ist in Form eines Programmfensters umgesetzt, das mit Hilfe von Karteireitern in mehrere Seiten unterteilt ist. Diese sind mit „Suche“, „Kurzinfo“, „Publikationen“, „DBLP & CiteSeer“. „Projekte“ und „Lebenslauf“ betitelt. Als Startseite ist die unter dem Reiter „Suche“ befindliche Seite selektiert. In den anderen Seiten werden die Suchergebnisse präsentiert. Dementsprechend sind die zughörigen Karteireiter zum Zeitpunkt des Suchstarts und auch während der Suche inaktiv. Abbildung 2 zeigt diese Ansicht.

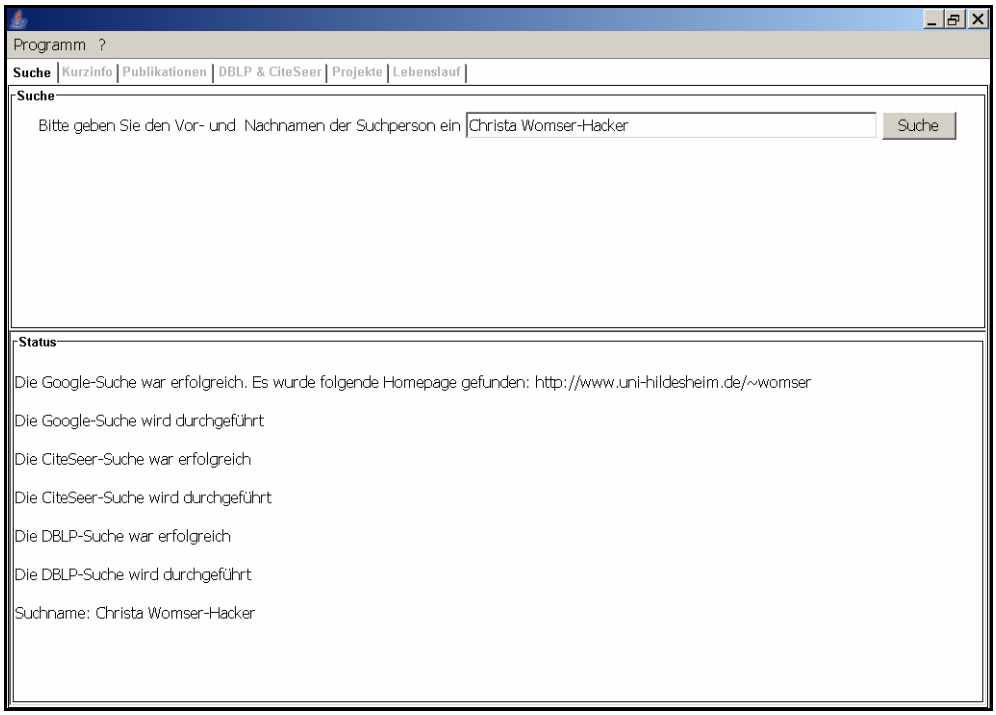

Abbildung 2: Such-Seite der Applikation

[27] http://www.eclipse.org/platform (verifiziert am 29.03.2006)

In dem Fenster befindet sich in der oberen Hälfte das Eingabefeld für den Suchnamen. In der unteren Hälfte ist das Status-Feld platziert. Mit den hier aufgelisteten Angaben, wird der Nutzer über den Fortschritt der Suche und den Erfolg der Teilsuchen informiert.

Nach erfolgreicher Suche werden die Seite „Kurzinfo“ geöffnet und die Karteireiter der Seiten aktiviert, zu denen Ergebnisse gefunden wurden. In der Seite „Kurzinfo“ werden der akademische Titel der Zielperson, die URL der Homepage, die E-Mail-Adresse und das Foto angezeigt, falls diese Informationen gefunden wurden, wie in Abbildung 3 zu sehen ist.

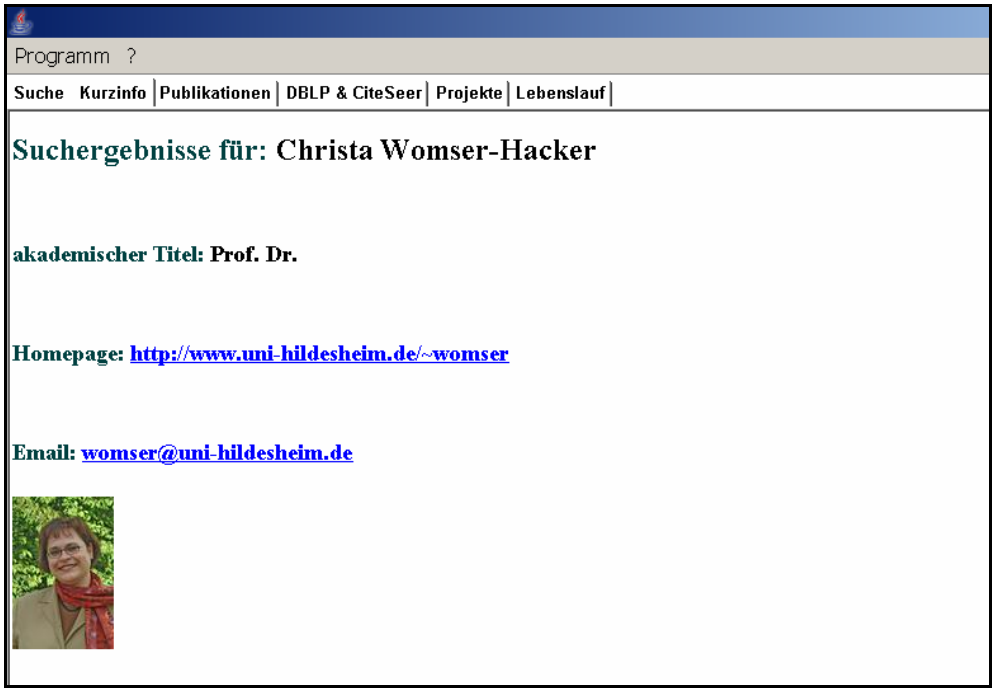

Abbildung 3: Kurzinfo-Seite für Suche nach "Christa Womser-Hacker" (06.03.2006)

Die dem Karteireiter „Publikationen“ zugeordnete Seite enthält die Publikationsangaben, die aus der Homepage der Zielperson extrahiert wurden. Die Seite „DBLP & CiteSeer“ zeigt die in den beiden Quellsystemen DBLP und CiteSeer gefundenen Publikationen, wobei die Ergebnisse nach Fundort unterteilt aufgelistet sind. Veröffentlichungen, die in beiden Quellen gefunden wurden, sind farblich gekennzeichnet. Abbildung 4 zeigt die DBLP & CiteSeer-Seite für die Zielperson „René Schneider“. Die Seite „Projekte“ enthält die Angaben zu Projekten, die in der Homepage der Zielperson gefunden wurden und die Seite „Lebenslauf“ stellt den aus der Homepage extrahierten Lebenslauf dar.

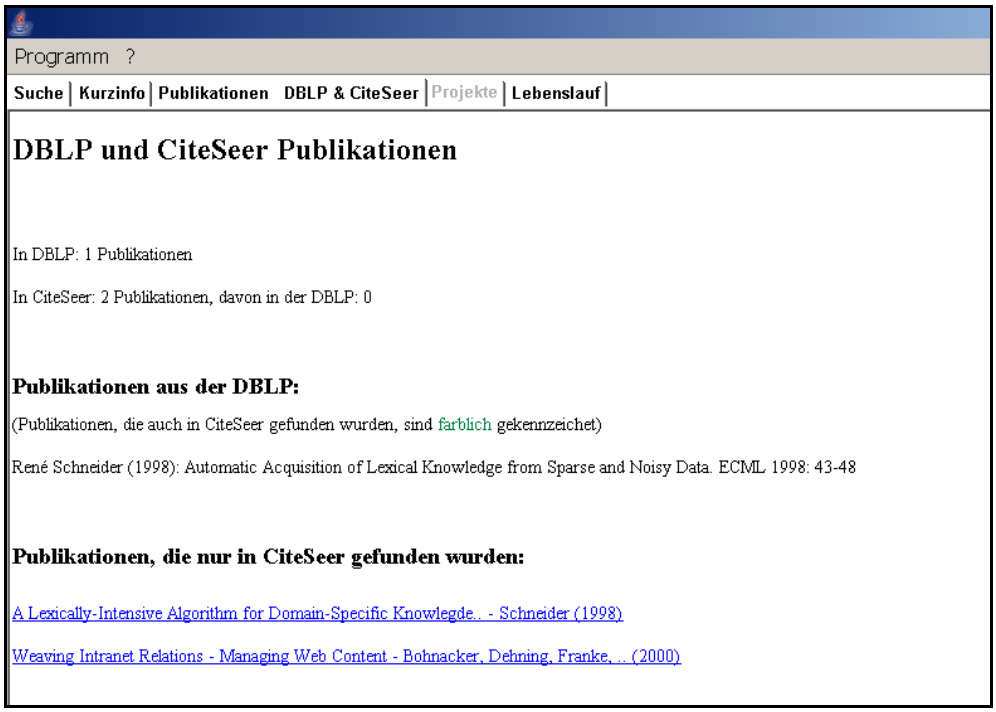

Abbildung 4: DBLP & CiteSeer-Seite (Suchname René Schneider, 06.03.2006)

Alle in den verschiedenen Ergebnisseiten angezeigten Links sind aktiv, so dass die verlinkte Seite bei Auswahl eines Links automatisch mit dem Standardbrowser des Betriebssystems geöffnet wird. So wird dem Benutzer die Navigation aus der Anwendung heraus ermöglicht. Die in den Quellseiten enthaltenen Links werden zusammen mit dem Text extrahiert und in der Ausgabe angezeigt, da diese oft zusätzliche Informationen enthalten, die für den Nutzer von Interesse sind. Dabei handelt es sich z. B. um Verweise auf die Volltexte der Publikationen oder auf detailliertere Projektseiten.

Generell ist es möglich, die angezeigten Informationen über eine entsprechende Menü-Funktion zu speichern. Das System generiert dazu eine HTML-Seite, die alle gefundenen Ergebnisse in der Form, in der sie in der Oberfläche dargestellt werden, enthält. So besteht die Möglichkeit, die Ergebnisse auch offline zu nutzen und auf Dauer zur Wiederverwendung zu archivieren.

4.2 Aufbau und Funktionsweise des Programms

Das Programm ist analog zu den in Kapitel 3.3 beschriebenen Teilschritten des Suchverfahrens in Packages unterteilt. Ein Package fasst jeweils die zu einer Teilaufgabe gehörenden Klassen zusammen. Im Folgenden wird auch von Modulen gesprochen.

Die erstellten Packages sind folgende:

- *useCiteSeer*: Die Klassen dieses Packages enthalten die Methoden zur Suche in dem CiteSeer-Index und der Extraktion der Veröffentlichungseinträge aus den Ergebnisseiten.

- *useDBLP*: Dieses Package enthält die Klassen, die die Einbindung des DBLP-Servers als Quelle umsetzen. Hier werden die Verbindungen zu den jeweiligen Autorenseiten hergestellt und die Veröffentlichungen der Zielperson extrahiert.

- *useGoogle*: Die hier zusammengefassten Klassen enthalten die Methoden zur Verwendung der Google-Suchfunktion. Außerdem setzen die Methoden die Bewertung der einzelnen Ergebnisseiten sowie das Herausfiltern der Homepage der Zielperson um.

- *evaluateHomepage*: Dieses Package enthält die Klassen, die das Lokalisieren und die Extraktion der Zielinformationen aus der jeweiligen Homepage zur Aufgabe haben.

- *interfaceSupport*: Die Klassen dieses Packages setzen die Zusammenstellung der Informationsitems für die Darstellung im Ausgabefenster um. Außerdem wird hier die Integration der Veröffentlichungen aus der DBLP und aus CiteSeer durchgeführt. Darüberhinaus beinhaltet das Package Klassen, die die Funktionalität der Oberfläche unterstützen, für das Suchverfahren selbst aber ohne Bedeutung sind.

Über die in diesen Packages gruppierten Klassen hinaus gibt es die Klassen *flowControl* und *GUI*. Erstere steuert den Gesamtablauf und Informationsfluss des Programms und fungiert als Schnittstelle zwischen den einzelnen Modulen. Die *GUI*-Klasse ist dem Namen entsprechend für die Oberfläche und die Interaktion mit dem Nutzer zuständig.

Abbildung 5 zeigt einen Überblick über die einzelnen Packages und Klassen und den Informationsfluss zwischen diesen. Im Folgenden wird die Arbeitsweise der einzelnen Packages und Klassen näher beschrieben. Hierbei liegt das Hauptaugenmerk auf der Umsetzung der in Kapitel 3 hergeleiteten Such- und Extraktionsverfahren und den dabei auftretenden Besonderheiten.

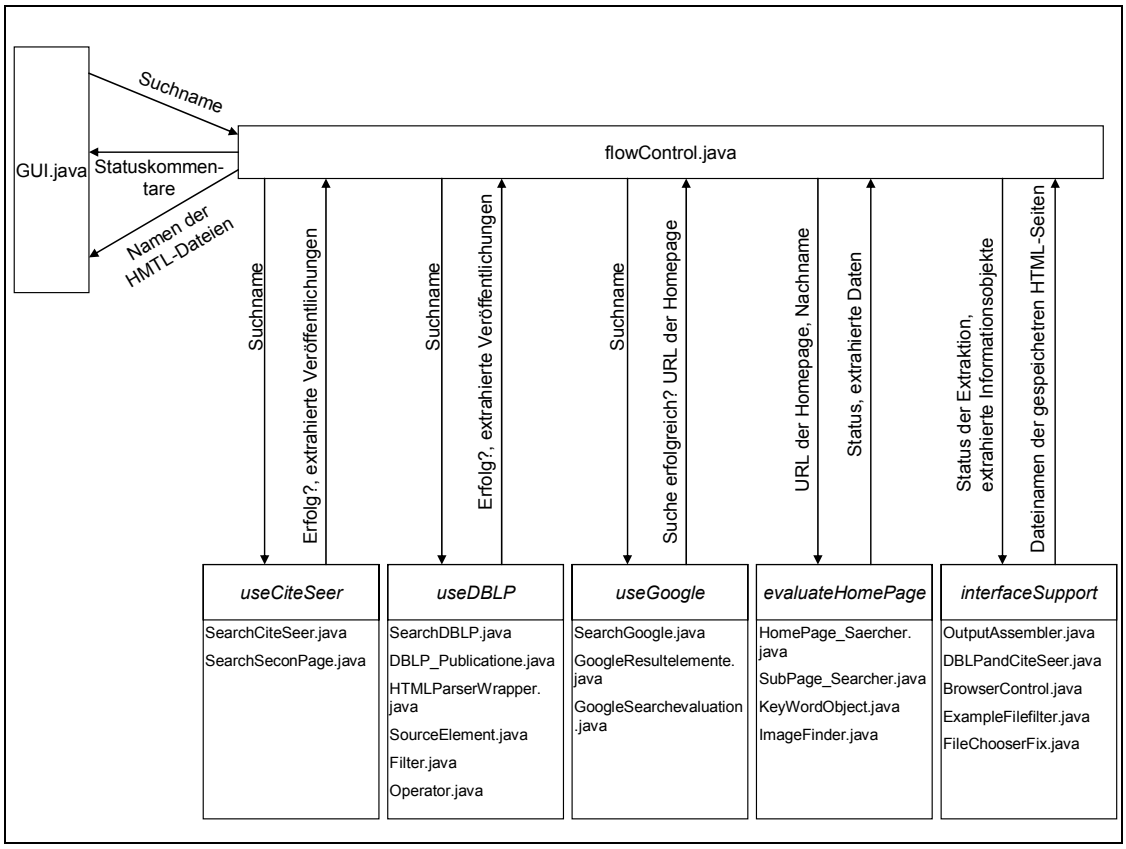

Abbildung 5: Struktur des Programms

4.2.1 UseCiteSeer-Modul

Die Aufgabe des CiteSeer-Moduls gliedert sich in zwei Teilaufgaben. Zunächst wird mit dem Namen der Zielperson die URL der Ergebnisseite generiert und der Quelltext dieser Seite geladen. Im nächsten Schritt werden die Ergebnisseite geparst und die Zielinformationen lokalisiert und extrahiert. Diese Funktionalität ist in der Klasse *SearchCiteSeer* implementiert. Wenn von einer Person mehr als 50 Veröffentlichungen im CiteSeer-Index gefunden werden, wird die Liste der Veröffentlichungen auf mehrere Ergebnisseiten aufgeteilt. Zur Extraktion der Titel der auf den weiteren Seiten angegebenen Einträge wird die Klasse *SearchSecondPage* eingesetzt.

Generierung der URL

Die URL einer Ergebnisseite für die Suche mit dem Namen der Zielperson als Suchterm, wobei die Suche auf den Header der Dokumente eingeschränkt wird, ist folgendermaßen[28]:

> http://citeseer.comp.nus.edu.sg/cs?q=***Vorname***%20***Nachname***&cs=1&submit=Search+Documents&af=Header&ao=Citations&am=50

Bestehen Vor- und oder Nachname aus mehreren Tokens, so werden diese durch die Zeichenfolge %20 voneinander getrennt angegeben. Alle im Namen vorkommenden Sonderzeichen müssen vor der URL-Generierung durch die entsprechenden Ersatzzeichen ersetzt werden, die sich aus dem Prozentzeichen % und einem zwei Zeichen umfassenden Hexadezimal-Code aus dem Standard ASCII-Zeichensatz zusammensetzen.

Wie in 3.2.1.1 beschrieben listet die Ergebnisseite von CiteSeer Zusammenfassungen der einzelnen Veröffentlichungen auf, welche u.a. aus dem Titel der Veröffentlichungen oder einem Teil des Titels, den Autorennamen und der Anzahl der Zitationen bestehen. Die detaillierten Angaben zu jeder Veröffentlichung finden sich jeweils auf einer Extraseite, zu der der String bestehend aus Titel- und Autorenangaben verlinkt ist. Diese Links auf die Veröffentlichungsseiten sind eindeutig identifizierbar, da sie die einzigen Verweise auf HTML-Seiten in den Ergebnisseiten sind. Somit wird die Endung „.html“ ausgenutzt, um die Titelstrings zu lokalisieren und als Label der Verweise zu extrahieren. Falls die Anzahl der Zitationen zu einer Veröffentlichung angegeben ist, so fungiert diese gleichzeitig als Link auf eine entsprechende Unterseite, welche die einzelnen Werke, in denen die Publikation referenziert wird, auflistet. Diese Links enthalten alle die Zeichenkette „context“, so dass sie eindeutig identifizierbar sind. Die Zitationsangaben werden als Label dieser Links extrahiert.

Wie in 3.2.1.1 beschrieben, wird, wenn die boolesche Suche mit der *AND*-Verknüpfung der Anfragebestandteile zu keinem Ergebnis geführt hat, automatisch diese Verknüpfung aufgehoben und eine erneute Suche durchgeführt. Da nur Veröffentlichungen, die den vollständigen Namen enthalten, eindeutig der Zielperson zuzuschreiben sind, muss das

[28] Da der Server von CiteSeer an der *Penn State's School of Informationen and Technology* während der Entwicklungsphase sehr instabil war, wurde der CiteSeer-Mirror an der *National University of Singapore* verwendet (http://citeseer.comp.nus.edu.sg/cs).

Suchverfahren erkennen, wenn eine Suche mit dem Namen als Phrase erfolglos war und dem Nutzer mitteilen, dass in dem CiteSeer-Index keine Einträge zu der Zielperson gefunden wurden. In Abbildung 6 ist zu sehen, dass die Ergebnisseite von CiteSeer in dem Fall den Nutzer mit dem farblich gekennzeichneten Hinweis „No documents match Boolean query. Trying non-Boolean relevance query.“ informiert.

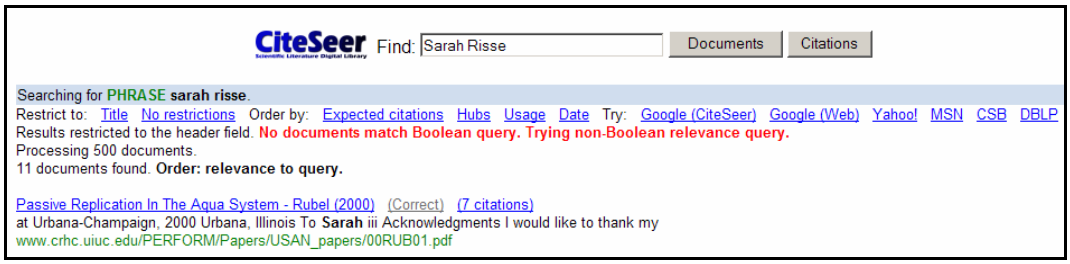

Abbildung 6: Fehlermeldung in CiteSeer

Die Ergebnisseite ist ein HTML-Dokument und dementsprechend sind die beschriebenen Struktur- und Layoutinformationen über die verwendeten HTML-Tags spezifiziert. Zur Nutzung dieser Informationen und zum Lokalisieren der Zielinformationen wird dementsprechend ein HTML-Parser verwendet, um die HTML-Dokumente zu verarbeiten. In dem entwickelten Programm wird der in der Java Klassenbibliothek enthaltene *HTMLEditorkit.Parser* benutzt. Dieser Parser wird auch in den anderen Modulen zur Analyse der HTML-Dokumente eingesetzt und spielt in den Extraktionsmethoden eine zentrale Rolle. Daher wird seine Funktion kurz erläutert.

HTML-Parser

Der Parser analysiert ein HTML-Dokument und reagiert auf jeden Tag und jedes Textelement: Der Parser benötigt ein Objekt der Klasse *HTMLEditorKit.ParserCallback* als Handler für die HTML-Tags und Textelemente. Jedes Mal, wenn der Parser auf einen Tag oder einen Textelement stößt, ruft er die entsprechende Methode des *ParserCallback*-Objekts auf. Als Parameter übergibt der Parser jeweils den Tag-Namen und die Attribute des Tags, bzw. die gelesene Zeichenkette und die aktuelle Position im Quelldokument. Entsprechend der Klassifizierung der HTML-Tags in Start-Tags, End-Tags, Simple-Tags, Empty-Tags und Textelemente beinhaltet die *ParserCallback*-Klasse die folgenden Methoden:

- *public void handleStartTag (HTML.Tag t, MutableAttributSet a, int pos)*
- *public void handleEndTag (HTML.Tag t, int pos)*

- *public void handleSimpleTag (HTML.Tag t, MutableAttributSet a, int pos)*
- *public void handleEmptyTag (HTML.Tag t, MutableAttributSet a, int pos)*
- *public void handleText (char[] data, int pos)*

Die Klassen, die die Informationsextraktion aus den HTML-Seiten umsetzen, sind Unterklassen der *ParserCallback*-Klasse, so dass diese Methoden geerbt und der Umsetzung der jeweiligen Extraktionsregeln entsprechend überschrieben werden.

Extraktion der Zielinformationen

Vor der Extraktion der Veröffentlichungen wird zunächst überprüft, ob die aufgelisteten Einträge das Resultat einer Suche mit dem vollständigen Namen sind, in dem der Quelltext nach der in Abbildung 6 sichtbaren Fehlermeldung durchsucht wird. Diese ist an der roten Schriftfarbe zu erkennen, die im Fall der CiteSeer-Ergebnisseite über das HTML-Attribut *color = red* im Font-Tag umgesetzt ist.

Anschließend werden unter Verwendung der *handleStartTag*- und *handleText*-Methoden alle Verweis-Tags gesucht, deren *href*-Attribut den String „.html“ enthalten. Die Links und ihr Label, also der Titel-Autoren-String der jeweiligen Veröffentlichung werden extrahiert. Wird ein Verweis-Tag mit einem *href*-Attribut gefunden, das den String „context“ enthält, wird das Label dieses Links als Zitationsangabe dem vorangegangenen Titel zugeordnet. Wird ein Link mit dem Label „Next 50“ gefunden, wird die zugehörige URL extrahiert und ein Objekt der Klasse *SearchSecondPage* mit der URL als Parameter aufgerufen. Da für den späteren Abgleich mit den in der DBLP enthaltenen Veröffentlichungen der Titel der einzelnen Publikationen benötigt wird, werden die Label in Titel- und Autoren-Strings zerlegt.

Das Ergebnis der Extraktion ist also eine Liste der Veröffentlichungseinträge zusammengesetzt aus dem Linklabel, bestehend aus Titelteilen und den Autorennamen, dem Titel, der URL der Veröffentlichungsseite und, falls vorhanden, der Zitationsangabe.

Abbildung 7 fasst den Ablauf im CiteSeer-Modul zusammen.

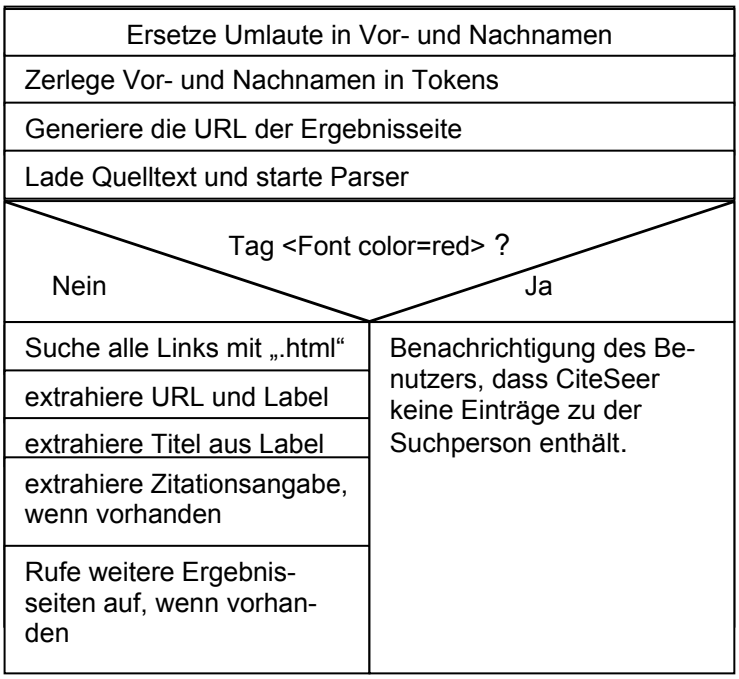

Abbildung 7: Ablaufdiagramm für das CiteSeer-Modul

4.2.2 UseDBLP-Modul

Analog zu dem CiteSeer-Modul lässt dich die Aufgabe des DBLP-Moduls in zwei Teilaufgaben unterteilen. In der ersten wird die URL der Autorenseite der Zielperson generiert und der Quelltext der Seite geladen. Die zweite Aufgabe umfasst das Lokalisieren der einzelnen Publikationseinträge und die Identifizierung der einzelnen Veröffentlichungsbestandteile. Im Folgenden wird beschrieben, wie diese Aufgaben im Detail realisiert sind und welche Besonderheiten zu beachten sind.

URL-Generierung

Die URL jeder Autorenseite der DBLP ist der Pfad dieser Seite im Author-Tree des DBLP-Servers. Sie setzt sich jeweils aus einem konstanten Teil und einem namensspezifischen Teil zusammen:

> http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/**Anfangsbuchstabe des Nachamens** / **Nachname**:**Vorname**.html

Um diese URL für die Autorenseite der jeweiligen Zielperson zu generieren, wird der Suchname in Vor- und Nachnamen unterteilt und vorhandene Sonderzeichen durch entsprechende Ersatzzeichen ersetzt. Die in der Bezeichnung der Pfade in der DBLP ver-

wendeten Ersatzzeichenketten für diese Sonderzeichen orientieren sich an den in der HTML-Syntax verwendeten benannten Zeichen.

Nicht zu jeder Zielperson ist eine Autorenseite in dem DBLP-Server vorhanden, so dass nach der URL-Generierung zunächst festgestellt werden muss, ob die generierte URL existiert. Das Vorhandensein der URL wird geprüft, indem versucht wird, eine Verbindung zu der Seite herzustellen und der Antwortcode des Servers überprüft wird. Dieser ist für eine nicht gefundene Seite *404*. Besteht die generierte URL nicht, geht das System davon aus, dass zu der Zielperson kein Eintrag in der DBLP besteht. Existiert die Autorenseite der Zielperson, so wird im nächsten Schritt der Inhalt der Seite geladen, der in Form von HTML-Quellcode vorliegt.

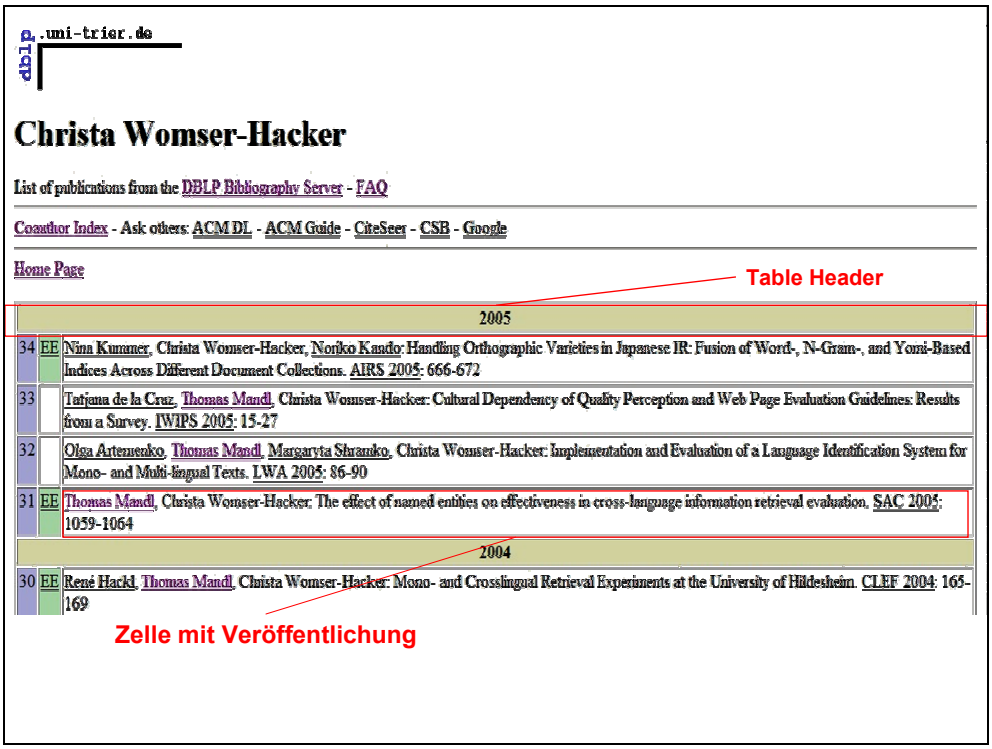

Abbildung 8: Ausschnitt aus einer Autorenseite der DBLP (Stand 27.02.2006)

Wie in Kapitel 3.2.1.3 dargestellt, wird die Struktur der Autorenseite dazu verwendet, die zu extrahierenden Veröffentlichungseinträge zu lokalisieren. Abbildung 8 zeigt eine Autorenseite der DBLP, deren Aufbau sich folgendermaßen beschreiben lässt: Die Veröffentlichungen sind in einer Tabelle aufgelistet, deren Inhalte chronologisch geordnet sind, so dass die aktuellsten Werke als erstes aufgeführt werden. Die Tabelle ist über Tabellenkopfzeilen („Table Header“) nach Jahren unterteilt. Die Publikationsangaben werden immer in der dritten Zelle einer Zeile aufgeführt, wobei jeweils eine Zelle die Angaben zu einer Veröffentlichung enthält. So sind die Veröffentlichungseinträge ein-

deutig im Quelltext zu lokalisieren und werden wie folgt extrahiert. Die Extraktionsmethode extrahiert die Inhalte der dritten Zelle jeder Zeile zwischen den einzelnen Tabellenkopfzeilen. So kann auch zu jeder Veröffentlichung das Veröffentlichungsjahr eindeutig als der Inhalt der vorangehenden Tabellenkopfzeile bestimmt werden.

Zur Umsetzung dieses Extraktionsverfahrens wird mit Hilfe des HTML-Parsers und der in Luton [2002] bereitgestellten Klassen (*HTMLParserWrapper.java*, *SourceElement.java*, *Operator.java*, *Filter.java*) ein Baum der Quellseite erstellt, der nur die Strukturtags und ihre Attribute berücksichtigt. Abbildung 9 zeigt einen Ausschnitt aus diesem Baum. Die zu extrahierenden Informationen sind fett markiert. Jedes einzelne Inhaltselement ist über den Key adressierbar und somit der Inhalt jeder Tabellenzelle direkt lokalisierbar. Die Umsetzung dieser Extraktionsmethode ist die Aufgabe der Klasse *SearchDBLP*.

```
.table[0].@border[0]=1
.table[0].tr[0].th[0].@colspan[0]=3
.table[0].tr[0].th[0].@bgcolor[0]=#FFFFCC
.table[0].tr[0].th[0].text[0]=2005
.table[0].tr[1].td[0].@valign[0]=top
.table[0].tr[1].td[0].@bgcolor[0]=#CCCCFF
.table[0].tr[1].td[0].@align[0]=right
.table[0].tr[1].td[0].a[0].@href[0]=http://dblp.uni-
trier.de/rec/bibtex/conf/airs/KummerWK05
.table[0].tr[1].td[0].a[0].@name[0]=p35
.table[0].tr[1].td[0].a[0].text[0]=35
.table[0].tr[1].td[1].@valign[0]=top
.table[0].tr[1].td[1].@bgcolor[0]=CCFFCC
.table[0].tr[1].td[1].a[0].@href[0]=http://dx.doi.org/10.1
007/11562382_65
.table[0].tr[1].td[1].a[0].text[0]=EE
.table[0].tr[1].td[2].a[0].@href[0]=http://www.informatik.
uni-trier.de/~ley/db/indices/a-tree/k/Kummer:Nina.html
.table[0].tr[1].td[2].a[0].text[0]=Nina Kummer
.table[0].tr[1].td[2].text[0]=, Christa Womser-Hacker,
.table[0].tr[1].td[2].a[1].@href[0]=http://www.informatik.
uni-trier.de/~ley/db/indices/a-tree/k/Kando:Noriko.html
.table[0].tr[1].td[2].a[1].text[0]=Noriko Kando
.table[0].tr[1].td[2].text[1]=: Handling Orthographic Va-
      rieties in Japanese IR: Fusion of Word-, N-Gram-,
      and Yomi-Based Indices Across Different Document
      Collections.
.table[0].tr[1].td[2].a[2].@href[0]=http://www.informatik.
uni-trier.de/~ley/db/conf/airs/airs2005.html#KummerWK05
.table[0].tr[1].td[2].a[2].text[0]=AIRS 2005
```

Abbildung 9: Ausschnitt aus dem Strukturbaum für die Autorenseite von Christa Womser-Hacker (http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/w/Womser=Hacker:Christa.html)

Extraktion der Veröffentlichungsbestandteile

Aus jeder extrahierten Veröffentlichung und dem zugehörigen Veröffentlichungsjahr wird ein Objekt der Klasse *DBLP_Publication* generiert, in welcher der Veröffentlichungsstring in seine einzelnen Bestandteile zerlegt und diese als Attribute dieser Klasse gespeichert werden. Dazu wird der einheitliche Aufbau der Veröffentlichungseinträge und die einheitliche Verwendung von Trennzeichen ausgenutzt: Ein Publikationseintrag setzt sich aus den Namen des Autors oder der Autoren, dem Titel, der Bezeichnung des Veröffentlichungsmediums, bzw. dem Konferenznamen und, falls vorhanden, den Seitenzahlen zusammen. Die Autorennamen sind durch Kommata voneinander getrennt und setzen sich aus Vor- und Nachnamen zusammen. Nach dem letzten Autoreneintrag folgt ein Doppelpunkt und nach dem Titel der Veröffentlichung ein Satzzeichen, in der Regel ein Punkt. Die Seitenzahlen folgen auf einen Doppelpunkt nach dem Veröffentlichungsmedium (siehe Abbildung 8).

Das Ergebnis der Extraktion aus den Autorenseiten des DBLP-Servers ist also eine Menge von *DBLP_Publication*-Objekten, die jeweils die einzelnen Bestandteile eines Publikationseintrags enthalten.

Abbildung 10 fasst den Ablauf der Informationsextraktion aus einer Autorenseite zusammen.

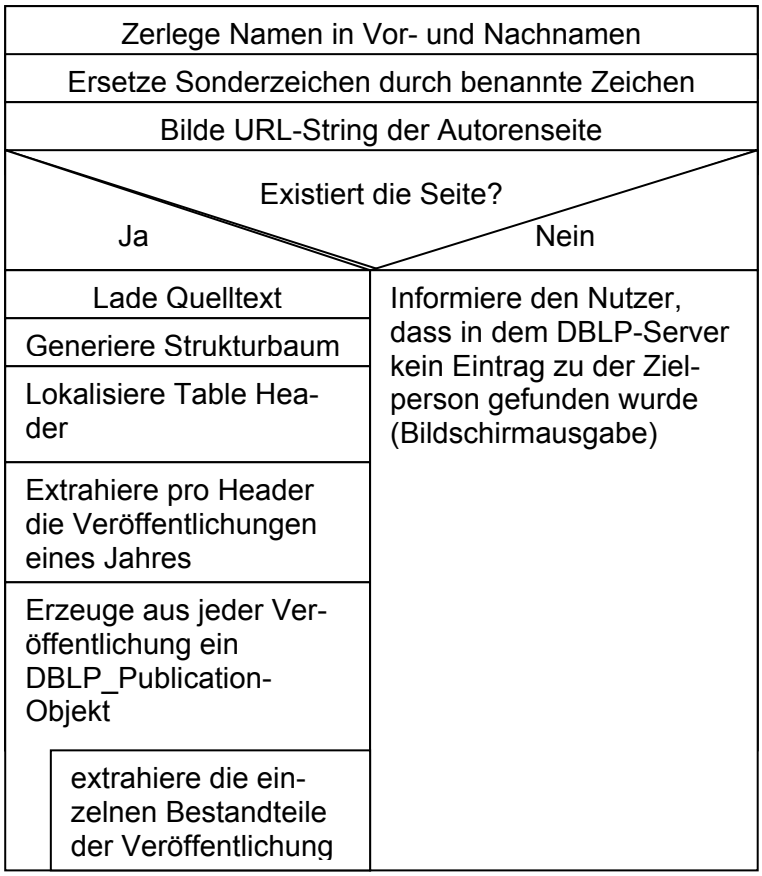

Abbildung 10 Ablaufdiagramm für das DBLP-Modul

4.2.3 UseGoogle-Modul

Das *useGoogle*-Package umfasst die Klassen zur Durchführung der Google-Suche nach der Homepage der Zielperson, der Bewertung der Suchergebnisse sowie der anschließenden Auswahl der Homepage.

Zur Durchführung der Suche bietet Google die Verwendung seiner Funktionen als Web Service an, bei dem über eine SOAP-Schnittstelle die Suchanfragen an den Google-Server gesendet werden. Dieser führt die Suchanfragen aus und sendet das Ergebnis an den Client zurück. Google stellt zudem eine Java-Klassenbibliothek zur Verfügung, die ein Wrapper für diese SOAP-Schnittstelle ist. Der Aufbau dieser Klassenbibliothek soll kurz erläutert werden, um darauf aufbauend die Implementierung der Suche zu beschreiben[29] [30].

[29] Eine detailliertere Spezifikation findet sich in der zugehörigen JavaDoc, die zusammen mit der Klassenbibliothek auf der Seite http://www.google.com/apis/download.html (verifiziert am 29.03.2006) heruntergeladen werden kann.

[30] Eine ausführliche Erläuterung der Funktionen des WebService findet sich in der zugehörigen Anleitung unter http://www.google.com/apis/reference.html. (verifiziert am 29.03.2006)

Die Klasse *GoogleSearch* dient dazu, die benötigten SOAP-Objekte und den Client zu erzeugen sowie die Verbindung zu dem Server herzustellen. Dies wird durch den Konstruktor der Klasse umgesetzt. Zur Nutzung der Dienste des Google Web Service ist eine Lizenznummer (engl. *Licence Key*) nötig. Um eine solche Lizenznummer zu erhalten, muss ein Google-Account angelegt werden[31]. Mit dieser Lizenznummer ist es dem Inhaber erlaubt, täglich 1000 Anfragen an den Google-Server zu stellen. Diese Anfragen können nicht nur Suchanfragen sein, es kann auch die Rechtschreibprüfungs-Funktion von Google genutzt werden. Die Lizenznummer wird dem Server mit der Methode *setKey(licencekey)* übermittelt.

Weiterhin können Einschränkungen in der Suche vorgenommen werden, wie es auch in der Online-Suche möglich ist. Zu diesen gehören das Herausfiltern von nicht-jugendfreien Seiten (*setSafeSearch(true)*), sowie die Einschränkung der Sprache (*setLanguageRestrict(Sprachenkürzel)*) oder des Herkunftslandes der Ergebnisdokumente (*setRestrict(countrycode)*).

Der Anfrageterm wird mit der Methode *setQueryString(Anfrageterm)* übermittelt und die Suche mit der Methode *doSearch()* ausgeführt.

Als Ergebnis der Suche wird dem Client ein Objekt der Klasse *GoogleSearchResult* übergeben. Dieses beinhaltet ein Array mit den einzelnen Ergebnissen der Suche. Pro Anfrage erhält der Client maximal zehn Resultate. Wünscht der Nutzer mehr Ergebnisse, so muss er eine weitere Anfrage stellen und spezifizieren, von welcher Rankingposition ausgehend die nächsten zehn Ergebnisse ermittelt werden sollen. Jedes der Suchergebnisse ist in einem Objekt der Klasse *GoogleSearchResultElement* gespeichert, welches als Attribute die einzelnen Bestandteile Titel, URL und Snippet des Suchergebnisses besitzt. Über *Getter*-Methoden kann auf diese Attribute zugegriffen werden. Die Rankingposition des einzelnen Ergebnisses ergibt sich aus der Position in dem Array der *GoogleSearchResultElement*-Objekte.

Das *useGoogle*-Package gruppiert die Klassen *SearchGoogle*, *GoogleResultElement* und *GoogleSearchEvaluation*. Die Klasse *SearchGoogle* führt die Suche in Google mit den vorgestellten Methoden durch. So wird ein Objekt der Klasse *GoogleSearch* erstellt, der Lizenzschlüssel übergeben und die Einschränkungen für die Suche gesetzt. Der Fil-

[31] https://www.google.com/accounts/NewAccount?continue=http://api.google.com/createkey&followup=http://api.google.com/createkey (verifiziert am 29.03.2006)

ter für nichtjugendfreie Seiten wird aktiviert und die Suche auf deutschsprachige Seiten eingeschränkt. Anschließend wird der Term für die Suchanfrage zusammengefügt. Er besteht aus dem eingegebenen Namen der Zielperson, mit Anführungszeichen versehen, damit die Suchterme als Phrase behandelt werden. Der Ausschluss von PDF-, PostScript und PowerPoint-Dokumenten wird mit dem Schlüsselwort *filetype* in der Suchanfrage angegeben. Das Template für den Suchstring ist demnach folgendes:

„“Vorname Nachname“ -filetype:pdf -filetype:ps -filetype:ppt”

Pro Suchergebnis wird ein Objekt der *GoogleResultElement*-Klasse erzeugt, dem das zugehörige *GoogleSearchResultElement*-Objekt und die Rankingposition als Parameter übergeben werden. Die Objekte der Klasse *GoogleResultElemen*t speichern für jedes Ergebniselement die von Google gelieferten Attribute sowie diejenigen, die in der Bewertung der einzelnen Ergebnisse gesetzt werden.

Die Klasse *GoogleSearchEvaluation* führt die Überprüfung der in 3.2.3.2 beschriebenen Eigenschaften, die in die Bewertung einfließen, und das Ranking der Ergebnisseiten durch. Mit Hilfe von String-Vergleichen werden zunächst die Ergebnisse ermittelt, die aus der DBLP, von CiteSeer oder aus der ACM Digital Library stammen. Der Titel von Ergebnisseiten, die aus der DBLP stammen, fängt mit dem String „DBLP“ an, die URLs der Ergebnisseiten von CiteSeer und der DBLP enthalten jeweils die URL des Servers, d. h. „citeseer.ist.edu" bzw. "portal.acm.org“. Die so identifizierten Ergebnisse werden aus der Menge der Ergebnisse gelöscht, da ausgeschlossen werden kann, dass es sich bei ihnen um die Homepage der Zielperson handelt.

Im nächsten Schritt werden die Ergebnisse ausgeschlossen, die den Nachnamen der Zielperson weder in der URL noch im Titel aufführen. In der URL wird auch das Vorkommen eines Teils des Nachnamens untersucht, so dass der Nachname in Trigramme zerlegt wird und jeweils pro Trigramm überprüft wird, ob es in der URL enthalten ist. Im Titel muss der vollständige Nachname enthalten sein.

Im folgenden Schritt wird mit Hilfe von Regulären Ausdrücken überprüft, ob die verbliebenen Ergebnisse die in 3.2.3.2 festgelegten Schlüsselwörter bzw. -zeichenketten enthalten. Mit dem Regulären Ausdruck "*Prof*\\. *Dr*\\.|*Prof Dr*|*Dr*|*Prof*\\.|*MA*|*Dipl*\\.-*Inf.*|*Dipl*\\. *Ing*\\.|*Dipl*\\. *Inf.*|*Dipl*\\.-*Ing*\\.|*M*\\.*A*\\." wird z.B. das Vorkommen eines akademischen Titels ermittelt. Nach der Ermittlung der Werte für die einzelnen zu bewertenden Eigenschaften der Suchergebnisse wird der Gesamtwert jeder Ergebnisseite berechnet, wobei jede zutreffende Eigenschaft mit einem Punkt gewertet wird. Die Sum-

me dieser Punkte wird gebildet und zusätzlich die Rankingposition in der Ergebnisliste von Google mit in die Bewertung einbezogen. Wie in Kapitel 3.2.3.2 beschrieben enthält das an erster Position gerankte Element einen Punkt, das an zweiter Position 0,9 Punkte usw., so dass die verwendete Berechnungsformel Folgende ist: (1 - (Google-Rank - 1) * 0.1).

Entsprechend der Heuristik zur Rangfolge akademischer Titel (siehe Kapitel 3.2.3.2), wurde eine Komparationsmethode implementiert, die den Vergleich akademischer Titel ermöglicht. Mit dieser werden Ergebnisseiten ausgeschlossen, die einen akademischen Titel enthalten, der in der Rangfolge vor dem höchsten akademischen Titel, der in den Seiten gefunden wurde, steht.

Aufbauend auf der Bewertung der einzelnen Ergebnisseiten wird ein Ranking der ermittelten Werte erstellt. Die URL des am höchsten gerankten *GoogleResultElement*-Objektes wird abgefragt und als Ergebnis der Evaluierung an das Objekt der *Search-Google*-Klasse zurückgegeben. Diese liefert diese URL an die Klasse *flowContro*l.

4.2.4 EvaluateHomepage-Modul

Die Klassen des *evaluateHomepage*-Package haben die Aufgabe, die Informationen in der Homepage der jeweiligen Zielperson zu lokalisieren und aus dieser zu extrahieren. Die Zielinformationen sind die E-Mail-Adresse, der akademische Titel, falls er während der Ermittlung der Bewertung nicht gefunden wurde, ein Foto der Zielperson, Angaben zu Publikationen und Projekten und der Lebenslauf. In dem Package gruppiert sind die Klassen *HomePage_Searcher*, *SubPageSearcher*, *KeyWordObject* und *ImageFinder*.

Die Hauptklasse ist die Klasse *HomePage_Searcher*. Hier wird zunächst eine Verbindung zu der in dem *useGoogle*-Package herausgefilterten URL der Homepage hergestellt, und der Quelltext der Seite wird geladen. Der HTML-Code wird mit Hilfe der Methoden der Java-Klassenbibliothek *Tidy*[32] auf Inkonsistenzen in der HTML-Struktur überprüft und gegebenenfalls repariert. Der bereinigte Quelltext wird mit dem weiter oben beschriebenen HTML-Parser analysiert. Der gesamte Quelltext wird in einen String eingelesen, wobei alle Tags, die nicht Struktur-Tags (Überschrift-, Absatz- oder Zeilenumbruch-Tags) oder Verweis-Tags sind, entfernt werden. Der Grund dafür ist,

[32] HTML Tidy ist eine freeware Software des W3C-Konsortiums, die automatisch Fehler in der HTML-Syntax bereinigt (vgl. http://www.w3.org/People/Raggett/tidy/ (verifiziert am 29.03.2006)). Die Java-Version wird unter http://sourceforge.net/projects/jtidy (verifiziert am 29.03.2006) zur Verfügung gestellt.

dass die Ergebnisse der Extraktion später als HTML-Seite im Ausgabefenster angezeigt werden und die Strukturierung der Ausgangsdokumente beibehalten werden soll. Denn die einzelnen Bestandteile der zusammenhängenden Informationen können in dem Prototyp nicht erkannt und somit nicht formatiert dargestellt werden.

Während des Parsens wird zudem ermittelt, ob die Stichwörter, die auf die Zielinformation hinweisen, in dem Text als Label eines Links oder als Textelement vorkommen. Die verwendeten Stichwörter sind „Projekt“, „Forschungsprojekt“, „Publikationen“, „Veröffentlichungen“, „Lebenslauf“, „Vita“ und „Zur Person“. Die Label aller Links werden mit einem Regulären Ausdruck auf enthaltene Schlüsselwörter überprüft. Wird so ein Label, das ein Schlüsselwort beinhaltet, gefunden, wird das Schlüsselwort mit dem kompletten Label und der zugehörigen URL extrahiert. Alle anderen Links werden zusammen mit ihrem Label gespeichert, um aus diesen im weiteren Verlauf den Verweis auf die E-Mail-Adresse herauszufiltern.

Ebenso wird jedes Textelement auf das Vorkommen eines der Schlüsselwörter überprüft. Wird ein Schlüsselwort gefunden, so wird ein Objekt der Klasse *KeyWordObject* erzeugt, das das Schlüsselwort und den vollständigen Textstring zusammen mit dem aktuellen Tag und seinen Attributen speichert. Der Tag und die Attribute geben die Formatierung dieses Textelementes an. So können im Folgenden alle Textelemente, die genauso formatiert sind wie das Schlüsselwort über einen Vergleich der Tags und ihrer Attribute identifiziert werden und ebenfalls als *KeyWordObject*-Objekt gespeichert werden.

Außerdem werden zu jedem gefundenen img-Tag der Speicherort der Bilddatei und der zugehörige Alternativtext gespeichert. Aus diesen wird im weiteren Verlauf das Foto der Zielperson extrahiert.

Nachdem der Quelltext geparst ist, wird für jede der Zielinformation festgestellt, ob ein zugehöriges Schlüsselwort als Titel-Attribut eines Objekts der *KeyWordObjects*-Klasse oder als Label eines Links vorliegt. Im ersten Fall wird die Position des Schlüsselworts in dem String, der den kompletten Seitentext enthält, bestimmt, sowie die Position des Titels des nächsten *KeyWordObjects*-Objekts. Als Zielinformation wird der Text zwischen diesen beiden Positionen extrahiert. Ist das Schlüsselwort ein Label, dann ist die Zielinformation auf der zugehörigen Unterseite zu finden. Es wird ein Objekt der Klasse *SubPage_Searcher* mit der URL als Parameter erzeugt. Dieses ruft die Unterseite auf und sucht das Schlüsselwort in dem Quelltext. Der auf das Schlüsselwort folgende Text

wird als Zielinformation extrahiert, wobei wie in der Behandlung der Hauptseite nur die Struktur-Tags und Links mitextrahiert werden.

Extraktion der E-Mail-Adresse

Ein Verweis auf eine E-Mail-Adresse wird in HTML durch den String „mailto“ eingeleitet. Dementsprechend werden zur Extraktion der E-Mail-Adresse der Zielperson die extrahierten Links daraufhin untersucht, ob sie diesen String enthalten. Wird eine E-Mail-Adresse gefunden, so wird überprüft, ob sie den Nachnamen oder einen Teil des Nachnamens der Zielperson enthält, um auszuschließen, dass die E-Mail-Adresse einer anderen Person extrahiert wird. Im nächsten Schritt werden eventuelle Zeichen, die als Spamschutz anstatt des @-Zeichens verwendet werden (z.B. (a) oder #), durch das @-Zeichen ersetzt. Im letzten Schritt wird die E-Mail-Adresse auf eine korrekte Syntax überprüft, wozu ein Regulärer Ausdruck verwendet wird.

Bild-Suche

Die Extraktion des Speicherorts des Fotos der Zielperson und das lokale Speichern dieser Bilddatei werden in der Klasse *ImageFinder* umgesetzt. Wie in Kapitel 3.2.2.3 beschrieben, wurde das Vorkommen des Nachnamens bzw. eines Schlüsselwortes im Dateinamen oder dem Alternativtext als charakteristisches Merkmal des Fotos der Zielperson heuristisch festgelegt.

Daher wird zunächst für jedes Bild der Dateiname aus dem vollständigen Speicherpfad extrahiert. Im nächsten Schritt wird untersucht, ob einer der Dateinamen den Nachnamen der Zielperson enthält. Wenn auf diese Weise keine Grafik als ein Foto der Zielperson identifiziert werden kann, werden die Alternativtexte der Bilder untersucht. Mit einem regulären Ausdruck wird das Vorkommen des Nachnamens oder eines der Schlagwörter „Bild“, „Foto“ oder „Photo“ überprüft. Falls ein Bild als das Foto der Zielperson identifiziert werden kann, wird die Bilddatei lokal gespeichert. Als Ergebnis der Suche wird bei Erfolg der Speicherpfad des Bildes zurückgegeben.

4.2.5 InterfaceSupport-Modul

Das Package *interfaceSupport* umfasst die Klassen *OutputAssembler* und *DBLPandCiteSeer*, die die extrahierten Informationen zusammenstellen und für die Ausgabe im Fenster der Applikation als HTML-Seiten speichern. Darüber hinaus enthält das Package die Klassen *BrowserControl*, *ExamapleFileFilter* und *FielChooserFix*, welche die Funktionen der Oberfläche unterstützen: Die Klasse *BrowserControl* öffnet einen ihr übergebenen Link im Standardbrowser des Systems. Sie wird aufgerufen, wenn der Nutzer in den Ausgabeseiten einen Link aktiviert. Die Klasse *ExampleFileFilter* dient dazu, dass in den Speicher-Dialogen nur HTML-Dateien angezeigt werden. Die Klasse *FileChooserFix* soll sicherstellen, dass der im Speicherdialog vorgegebene Dateiname bei einem Wechsel der Ordner nicht gelöscht wird.

Die Klasse *DPLBandCiteSeer* erstellt einen HTML-Quelltext, der die aus der DBLP und aus CiteSeer extrahierten Veröffentlichungen auflistet. Dazu wird aus jedem *DBLP_Publication*-Objekt, das jeweils eine der in der DBLP eingetragenen Veröffentlichungen umfasst, ein String erzeugt, der die Attribute dieses Objekts in Form einer Veröffentlichungsangabe enthält. Für jede dieser Veröffentlichungen wird überprüft, ob der Titel mit einem der Titel der aus CiteSeer extrahierten Publikationen übereinstimmt. Wenn das der Fall ist, wird der Titel mit HTML-Font-Tags umschlossen, die die farbliche Hervorhebung umsetzen. Falls zu der entsprechenden Veröffentlichung eine Zitationsangabe vorhanden ist, wird diese zu dem Veröffentlichungsstring hinzugefügt.

Die aus CiteSeer extrahierten Veröffentlichungen, die nicht in der DBLP gefunden wurden, werden an die anderen Veröffentlichungen angehängt. Zu jeder dieser Veröffentlichungen ist der Link auf die einzelne Veröffentlichungsseite in dem CiteSeer-Server, das Label dieses Links und gegebenenfalls die Zitationsangabe vorhanden. Diese Angaben werden zu einem String in HTML-Format zusammengefügt und mit dem Teilstring, der die Publikationen aus der DBLP enthält, verknüpft. Der Output dieser Klasse ist der Quelltext für die HTML-Seite, die die extrahierten Veröffentlichungen anzeigt, gespeichert in einem String.

In der Klasse *OutputAssembler* werden die Daten zur Anzeige im Ausgabefenster als HTML-Seite gespeichert. Dazu werden der Klasse die in den anderen Modulen extrahierten Informationen als Strings übergeben. Für die Seite „Kurzinfo“ wird aus dem Namen der Zielperson, dem akademischen Titel, der URL der Homepage, der E-Mail-Adresse und dem Bild der Quelltext für eine HTML-Seite generiert. Die genaue Umset-

zung dieses Vorgangs zeigt der Ausschnitt aus der Methode generateInfoOutput() in Abbildung 11. Dieser String, der den kompletten HTML-Code für die Seite enthält wird lokal in einer temporären HTML-Datei gespeichert. Die Datei wird so angelegt, dass sie, wenn das Programm geschlossen wird, automatisch gelöscht wird. Der Dateiname wird im weiteren Verlauf des Programms über die Klasse *flowControl* dem Fenster übergeben und als Inhalt des Karteireiters „Kurzinfo“ dargestellt.

Die Inhalte für die anderen Karteireiter werden auf dieselbe Art und Weise zusammengefügt und temporär als HTML-Seiten gespeichert. Die aus den Homepages extrahierten Informationen, also die Angaben zu Publikationen, Projekten und Lebenslauf, sind wie weiter oben beschrieben mit den Strukturtags der Quelldokumente versehen, so dass diese Strings nur noch mit einer Überschrift ergänzt werden. Der String mit den Publikationen aus CiteSeer und der DBLP wurde in der Klasse DBLPandCiteSeer mit den entsprechenden Format-Tags versehen.

```
public String generateInfoOutput() {

        String info = "<html> <h1> <font color=\"#004040\">Suchergebnisse für:
        </font> <b>" + name + "</b> </h1>";
                if (titleFound) {
                        info = info + "<p> <h2> <font color=\"#004040\"> akademischer
                        Titel: </font>" + title + "</h2></p>";
                        }

                info = info + "<p> <h2> <font color=\"#004040\"> Homepage: </font>
                <a href=\"" + hp_url + "\">" + hp_url + "</h2></p>";

                if (mailFound) {
                        info = info + "<p> <h2> <font color=\"#004040\"> Email: </font>
                        <a href=\mailto: + mail + "\">" + mail + "</a> </h2></p>";
                        }

                if (imgFound) {
                        info = info + "<p><img src=\"" + imgFile + "\"> </p>";
                        }

                info = info + "</html>";
                ...
```

Abbildung 11: Ausschnitt aus der Methode generateInfoOutput

Über die Methoden zum Speichern der einzelnen Informationen in HTML-Seiten hinaus, verfügt die Klasse *OutputAssembler* über eine Methode, die alle extrahierten Informationen in einem String zusammenfügt und als eine HTML-Seite speichert. Diese wird benötigt, wenn das Suchergebnis gespeichert werden soll. Die Klasse *flowControl* übergibt der Methode den vom Nutzer ausgewählten Speicherort und den Dateinamen.

Die Datei wird zusammen mit einem gleichnamigen Ordner, in dem das Bild abgespeichert wird, erzeugt.

5 Evaluierung der entwickelten Such- und Extraktionsverfahren

In diesem Kapitel wird die Evaluierung der einzelnen Komponenten des entwickelten Prototypen beschrieben. Dabei wird zunächst die Vorgehensweise vorgestellt und anschließend die Ergebnisse sowie daraus resultierende Verbesserungsmöglichkeiten erläutert.

5.1 Vorgehensweise

Um überprüfen zu können, ob das Verfahren zu einer Anfrage die richtigen Ergebnisse liefert, muss bekannt sein, welche der Informationen überhaupt zu der Zielperson im Web gefunden werden können. Daher wurde zunächst eine Testkollektion erstellt, die die Ergebnisse manueller Suchen zu einer Gruppe von Testpersonen umfasst.

Die Testpersonen wurden aus der Menge der Autoren ausgewählt, deren Beiträge in den Proceedings[33] des *9. Internationalen Symposiums für Informationswissenschaft* (ISI 2004) veröffentlicht worden sind. Von diesen wurden im ersten Schritt nur diejenigen Autoren ausgewählt, die im deutschsprachigen Raum tätig sind und nicht bereits in der Trainingsmenge enthalten sind. Für die verbleibenden 50 Personen wurde manuell nach den Homepages, den in diesen enthaltenen Zielinformationen und den Veröffentlichungseinträgen in der DBLP und CiteSeer gesucht. Bei der Suche nach den Homepages wurden diejenigen, welche hauptsächlich privaten Zwecken dienen und keine der Zielinformation enthalten, nicht als relevantes Ergebnis gewertet. Lediglich zu 26 der Personen konnte so eine Homepage gefunden werden. Diese 26 Personen wurden letztendlich in die Testkollektion aufgenommen.

Zur Evaluierung wurden die Ergebnisse der Suche mit dem Prototypen mit den Ergebnissen der manuellen Suche verglichen. Dabei wurden die Extraktion der Publikationseinträge aus der DBLP und CiteSeer, die spezialisierte Suche nach Homepages und die Extraktion der Zielinformationen aus den Homepages untersucht. Die Teilaufgaben der Suche nach den Homepages und der Informationsextraktion wurden separat evaluiert, da die Ergebnisse der Informationsextraktion davon abhängig sind, dass die vorhergehende Suche die richtige Homepage liefert. So wurden, um die Fehlerquellen eindeutig

[33] Bernard Bekavac, Josef Herget, Marc Rittberger (Hrsg.) (2004): Information zwischen Kultur und Marktwirtschaft - Proceedings des 9. Internationalen Symposiums für Informationswissenschaft (ISI 2004), Chur, 6. – 8. Oktober 2004. Hochschulverband für Informationswissenschaft.

identifizieren zu können, für die Evaluierung dieser Komponente die URLs der Homepages manuell eingegeben und der Schritt der Suche deaktiviert.

Bei der Evaluierung der Extraktion der Publikationseinträge aus der DBLP und CiteSeer werden die Fälle als Fehler gewertet, in denen Publikationseinträge, die nicht der Zielperson gehören, extrahiert oder aber keine Ergebnisse geliefert werden, obwohl Einträge der Zielperson vorhanden sind. Die Ergebnisse der Suche nach den Homepages werden als korrekt angesehen, wenn die gefundene Homepage mit der in der Testkollektion enthaltenen Homepage übereinstimmt. Wurden mit der manuellen Suche mehrere Homepages zu einer Zielperson gefunden wird das Ergebnis als korrekte gewertet, eine von diesen gefunden wurde. Als Fehler werden die Fälle gewertet, in denen keine Homepage gefunden wird oder aber eine Seite als Homepage identifiziert wird, die nicht mit der Homepage der Zielperson übereinstimmt. Als falsches Ergebnis der Extraktion der Zielinformationen aus den Homepages werden die Fälle gewertet, in denen ein vorhandenes Informationsobjekt nicht extrahiert wird oder ein falsches Informationsobjekt extrahiert wird, wie z.B. die E-Mail-Adresse oder der akademische Titel einer anderen Person.

Zur Evaluierung von Such- und Informationsextraktionsverfahren werden in der Informationswissenschaft am häufigsten die Evaluierungsmaße Recall und Precision verwendet [vgl. Ferber 2003, 86; Neumann 2001, 3]. Bei der Evaluierung eines Suchverfahrens gibt die Precision „den Anteil der relevanten Dokumente unter den gefundenen Dokumenten an“ und der Recall „gibt den Anteil der relevanten Dokumente an, die gefunden wurden“ [Ferber 2003, 87]. Die Berechnung der beiden Maße ist dementsprechend wie folgt:

$$\text{Precision} = \frac{\text{Anzahl gefundener relevanter Dokumente}}{\text{Anzahl gefundener Dokumente insgesamt}}$$

$$\text{Recall} = \frac{\text{Anzahl gefundener relevanter Dokumente}}{\text{Anzahl relevanter Dokumente insgesamt}}$$

Analog dazu gibt die Precision bei einem Informationsextraktionsverfahren den Anteil der korrekt gewonnenen Informationsobjekte an den insgesamt gewonnenen Informati-

onsobjekten an. Der Recall ist der Anteil der gewonnenen Informationsobjekte im Verhältnis zu den insgesamt gewinnbaren Informationsobjekten [vgl. Neumann 2001, 3].

Die Anwendung dieser quantitativen Evaluierungsmaße zur Bewertung des Prototypen stand angesichts der geringen Größe der Testkollektion und des frühen Entwicklungsstadiums des Verfahrens nicht im Vordergrund der Evaluierung. Trotzdem werden die Anzahl korrekter Ergebnisse und der Fehler angegeben, um eine erste Einschätzung der Qualität des Verfahrens zu ermöglichen. In erster Linie soll jedoch eine qualitative Evaluierung zeigen, ob der entwickelte Ansatz Erfolg versprechend ist oder nicht. Es sollen die Ursachen für falsche Ergebnisse analysiert werden, um darauf aufbauend Verbesserungsnotwendigkeiten bzw. -möglichkeiten aufzuzeigen. Darüber hinaus soll festgestellt werden, ob die ausgewählten Quellen und die ermittelten Heuristiken ausreichen, um die Zielinformationen zu finden und so das Informationsbedürfnis des Nutzers zu befriedigen.

5.2 Ergebnisse der Evaluierung

Informationsextraktion aus CiteSeer

Die Extraktion der Publikationsdaten aus CiteSeer liefert für 23 der 26 Testpersonen das richtige Ergebnis. Dabei sind aber lediglich zu drei der Testpersonen Veröffentlichungseinträge in CiteSeer enthalten. Zu den anderen Testpersonen liefert das Extraktionsverfahren das richtige Ergebnis, dass keine Veröffentlichungseinträge gefunden wurden.

Die falschen Ergebnisse sind auf zwei verschiedene Probleme zurückzuführen. Zum einen kann die Suche in CiteSeer nicht ausschließlich auf das Autorenfeld eingeschränkt werden, sondern nur auf den Header der Dokumente, der u.a. auch den Titel der Dokumente enthält. So ist es in einem Fall vorgekommen, dass der Nachname der Zielperson im Titel einer Veröffentlichung vorkommt und diese fälschlicherweise als Veröffentlichung der Zielperson extrahiert wurde. Dieser Fehler ist nicht auszuschließen, wenn nicht die einzelnen Veröffentlichungsseiten aufgerufen werden, um die exakten Autorenangaben zu identifizieren.

Darüber hinaus ist ein Fall der Namensambiguität aufgetreten, so dass Publikationen extrahiert worden sind, die nicht von der Zielperson, sondern von einem gleichnamigen Wissenschaftler stammen. Dieser Fehler ist schwer zu vermeiden. Eine mögliche Lösung wäre der Einsatz von Verfahren zur Namensdisambiguierung, die anhand der Titel-

stichwörter, Koautoren und Veröffentlichungsmedien entscheiden, ob mehrere Veröffentlichungen von ein und demselben Autor stammen. Ein solches Verfahren wird u.a. bei Han et al. [2004] vorgestellt.

Informationsextraktion aus der DBLP

Die Extraktion der Publikationsdaten aus der DBLP liefert mit einer Ausnahme zu allen Testpersonen die korrekten Ergebnisse. In dem Fall, in dem kein Ergebnis geliefert wird, obwohl ein Eintrag vorhanden ist, sind die Sonderzeichen im Namen nicht korrekt ersetzt worden, so dass die URL der Autorenseite falsch generiert wurde. Dieser Teil des Wrappers für den DBLP-Server muss dementsprechend angepasst werden.

Spezialisierte Suche nach Homepages

Die spezialisierte Suche nach den Homepages der Testpersonen liefert folgende Ergebnisse: Zu 13 der 26 Testpersonen wurde die richtige Homepage gefunden, zu neun Personen wurde keine Homepage gefunden und in vier Fällen wurden Seiten fälschlicherweise als Homepage identifiziert. Die Precision liegt also bei 0.76 und der Recall bei 0.5.

Eine Analyse dieser Ergebnisse zeigt, dass die Fälle, in denen das System keine Homepage finden konnte, auf die gleichen Ursachen zurückzuführen sind: Die ermittelten Heuristiken treffen nicht zu, so dass manche Homepages fälschlicherweise nicht erkannt oder von der Bewertung ausgeschlossen werden. In Folge dessen wurden weitere Untersuchungen durchgeführt, um zu überprüfen, ob ein niedrigerer Schwellenwert und der Verzicht auf das Ausschließen der Ergebnisse, in denen der Nachname nicht im Titel oder der URL vorkommt, dazu beiträgt, dass mehr Homepages korrekt identifiziert werden, also der Recall vergrößert werden kann.

Im zweiten Suchdurchgang wurde der Schwellenwert auf drei gesenkt, im dritten Versuch wurden keine Seiten vorab eliminiert und somit alle Seiten in die Bewertung einbezogen und im vierten Versuch die beiden Änderungen kombiniert angewandt. Tabelle 1 stellt die Ergebnisse der vier Suchläufe dar. Hier ist anzumerken, dass diese vergleichende Darstellung nur eine Tendenz wiedergeben kann, da die Ausgangssituation für die Bewertung in den verschiedenen Suchdurchgängen nicht gleich bleibend ist. Die Suchläufe sind an verschiedenen Tagen durchgeführt worden und es hat sich gezeigt,

dass Google auf dieselbe Anfrage in vielen Fällen nicht dieselben zehn Referenzen als Ergebnis liefert.

	Suche 1	Suche 2	Suche 3	Suche 4
Schwellenwert	4	3	4	4
Eliminierung der Ergebnisse ohne Nachnamen in URL oder Titel	Ja	Ja	Nein	Nein
Anzahl korrekte Ergebnisse	13	16	13	17
Anzahl falsche Ergebnisse	4	8	6	7
Anzahl kein Ergebnis	9	2	7	2
Precision	0.76	0.67	0.68	0.71
Recall	0.5	0.62	0.5	0.65

Tabelle 1: Ergebnisse der Untersuchung der spezialisierten Suche

Wird der Schwellenwert auf drei gesenkt, so erhöht sich wie erwartet die Anzahl der gefundenen Homepages und die Anzahl der Fälle, in denen kein Ergebnis gefunden wird, nimmt ab. Gleichzeitig steigt aber auch die Zahl der falschen Ergebnisse. Somit erhöht sich der Recall auf 0.62, die Precision sinkt allerdings gleichzeitig auf 0.67. Die Suche allein ohne Ausschluss der Ergebnisse, die den Nachnamen der Zielperson weder im Titel noch in der URL enthalten, führt zu keiner Verbesserung der Ergebnisse, es werden im Gegenteil mehr Seiten fälschlicherweise als Homepage identifiziert. Der Recall sinkt dementsprechend wieder auf 0.5 und die Precision auf 0.68. Die Kombination der beiden Änderungen liefert das beste Gesamtergebnis: Die Precision liegt bei 0.71 und der Recall verbessert sich auf 0.65 verbessert.

Das auftretende Phänomen, dass zwar die Anzahl der korrekten, gefundenen Ergebnisse erhöht werden kann, gleichzeitig aber auch die Anzahl der falschen Ergebnisse zunimmt, ist das aus dem Bereich des Information Retrieval bekannte Problem der Gegenläufigkeit von Recall und Precision [vgl. Ferber 2003, 87].

Über diese Untersuchung hinaus wurden die Fälle analysiert, in denen die Homepage nicht gefunden oder eine andere Seite fälschlicherweise als Homepage identifiziert wurde. Zum einen waren die Homepages der Zielpersonen nicht in der Menge der von Google gelieferten Ergebnisse enthalten. In diesen Fällen waren die Homepages aber auch nicht unter den ersten 50 Ergebnissen der Google-Onlinesuche. Eine weitere Ursache dafür, dass die Homepage nicht korrekt identifiziert wurde, war, dass die ermittelten typischen Eigenschaften von Homepages nicht zutrafen und folglich die Bewertung der Homepages niedrig war. Dies führte zunächst dazu, dass kein Ergebnis geliefert wurde.

Wurde zudem der Schwellenwert gesenkt, wurden andere Seiten fälschlicherweise als Homepage identifiziert. Außerdem ist der Fall aufgetreten, dass eine Person einen ebenfalls wissenschaftlich tätigen Namensvetter hat und dessen Homepage höher bewertet wurde. Diese Fälle sollen durch die positive Bewertung des Vorkommens von Begriffen und Abkürzungen der Informationswissenschaft in Titel, Snippet oder URL der Ergebnisse vermieden werden. In dem vorliegenden Fall trafen aber diese Eigenschaften auch nicht auf die Seite der Zielperson zu.

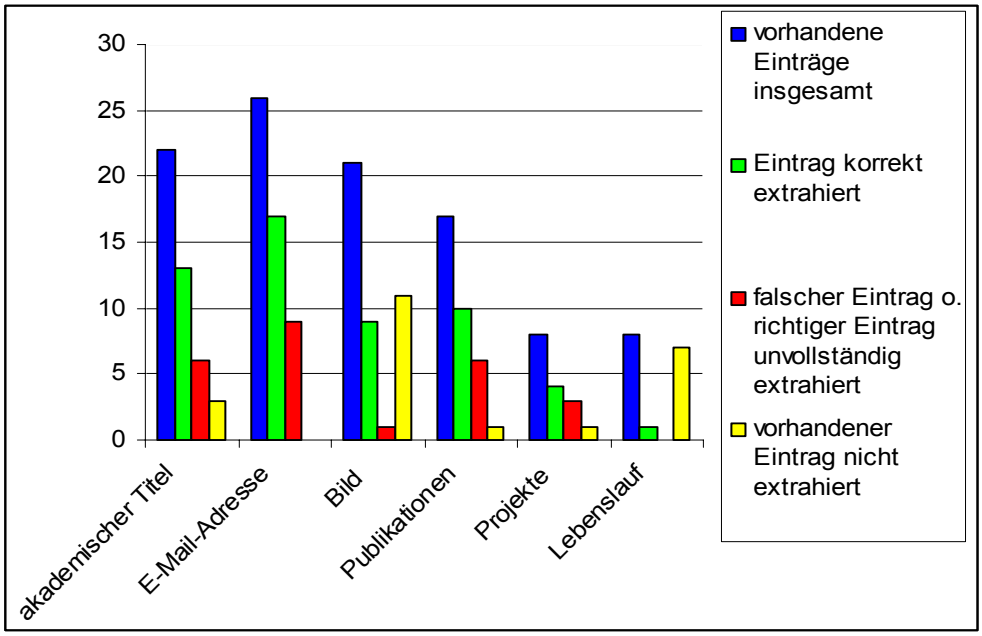

Abbildung 12: Ergebnisse der Informationsextraktion aus den Homepages

Informationsextraktion aus der Homepage

Während der Extraktion der Zielinformationen tritt bei einer Homepage ein nicht zu behebender Fehler auf[34], weswegen diese Seite für die Evaluierung dieser Teilaufgabe aus der Testkollektion entfernt wurde. Da aber für eine Testperson zwei relevante Homepages ermittelt wurden, die beide bei der Evaluierung berücksichtigt werden sollen, umfasst die Testkollektion weiterhin 26 Homepages. Im Folgenden werden die Ursachen für die Fälle, in denen falsche Informationen oder vorhandene Informationen nicht oder nur unvollständig extrahiert wurden, für die einzelnen Informationsobjekte analysiert. In Abbildung 12 sind die Ergebnisse der Evaluierung der Informationsextraktion aus den Homepages zusammengefasst.

[34] Der Fehler ruft eine *NullPointer-Exception* hervor, so dass überhaupt keine Ergebnisse angezeigt werden. Das Programm kann aber für weitere Suche verwendet werden.

Die Extraktion der akademischen Titel liefert für 13 der 22 Homepages, die einen akademischen Titel enthalten, das richtige Ergebnis. In fünf Homepages ist kein akademischer Titel angegeben. In einem dieser Fälle wird ein Titel extrahiert, der nicht der Titel der Zielperson ist. In den acht anderen Fällen wird entweder kein Titel oder der angegebene Titel nicht vollständig extrahiert, bzw. anstatt des Titels der Zielperson der akademische Titel einer anderen Person extrahiert.

Falsche Ergebnisse treten auf, wenn der Titel der Zielperson nicht angegeben ist, wohl aber zufällig eine andere Person mit akademischem Titel in der Homepage erwähnt wird, oder aber ein anderer akademischer Titel in dem Quelltext vor dem Titel der Zielperson aufgeführt ist. Um dies zu vermeiden, müsste die Extraktionsmethode die Nähe zu dem Namen der Zielperson berücksichtigen: Der akademische Titel wird in der Regel direkt vor dem Namen der Zielperson oder unmittelbar dahinter aufgeführt. So könnte man die Einschränkung vornehmen, dass ein akademischer Titel nur der Zielperson zugeschrieben wird, wenn er an einer dieser Positionen auftritt.

Drei andere Fehler sind darauf zurückzuführen, dass die vorab erstelle Liste akademischer Titel nicht umfassend genug ist. So werden die akademischen Titel der drei österreichischen Zielpersonen nicht vollständig extrahiert und ausgeschriebene Titel wie Diplom-Medienwissenschaftlerin nicht erkannt, da sie nicht durch den entsprechenden Regulären Ausdruck abgedeckt werden.

In 17 der 26 Testfälle wird die E-Mail-Adresse der Zielperson korrekt extrahiert. In den anderen Fällen werden die E-Mail-Adressen nicht extrahiert, obwohl sie in der Homepage angegeben sind. Es kommt nicht vor, dass eine falsche E-Mail-Adresse extrahiert wird.

In zwei der Homepages, aus denen die E-Mail-Adresse nicht extrahiert wird, ist diese nicht auf der Hauptseite aufgeführt sondern auf einer Unterseite mit dem Titel „Kontakt“. Die Unterseite ist von der Hauptseite verlinkt. Um auch in diesen Fällen die E-Mail-Adresse korrekt zu extrahieren, kann das Extraktionsverfahren folgendermaßen erweitert werden: Wenn auf der Hauptseite keine E-Mail-Adresse gefunden wird, muss nach einem Link mit dem Label „Kontakt“ gesucht, die verlinkte Unterseite aufgerufen und diese mit der vorhandenen Methode nach der E-Mail-Adresse durchsucht werden.

Eine weitere Ursache dafür, dass vorhandene E-Mail-Adressen nicht extrahiert werden, liegt darin, dass die Verweise bzw. die E-Mail-Adressen auf Grund von Spamschutzmaßnahmen für den Parser nicht als E-Mail-Adresse identifizierbar sind, wohl aber für den Nutzer, der die Homepage liest. In einem Fall ist das @-Zeichen nicht als Schriftzeichen sondern als Grafik eingefügt. Eine andere E-Mail-Adresse wird mit einer JavaScript Methode erst zur Laufzeit generiert. In einer weiteren Homepage ist die komplette E-Mail-Adresse als Grafik eingefügt und der Verweis ein PHP-Befehl. Diese Fehler sind folglich nicht zu vermeiden.

Des Weiteren können vorhandene E-Mail-Adressen bisher nicht erkannt werden, wenn diese nicht als E-Mail-Verweis sondern als reiner Text-String im Quelltext vorhanden sind. Dieses Problem tritt in zwei Fällen auf. Als Lösung müsste der Parser auch die Textelemente daraufhin überprüfen, ob sie eine gültige E-Mail-Adresse enthalten. Dazu kann dieselbe Methode verwendet werden, die die E-Mail-Verweise überprüft. In einem anderen Fall enthält die als Kontakt angegebene E-Mail-Adresse nicht den Namen der Zielperson und wird daher nicht extrahiert.

Darüber hinaus ist in einem Fall als Kontakt nicht eine E-Mail-Adresse angegeben, sondern zwei miteinander verknüpfte E-Mail-Adressen. Diese werden von der Syntaxüberprüfung als nicht korrekt eingestuft und folglich nicht extrahiert. Mit der weiter oben beschriebenen Erweiterung, auch die reinen Textelemente nach einer gültigen E-Mail-Adresse zu durchsuchen, würde die erste dieser beiden E-Mail-Adressen identifiziert werden.

Lediglich 21 der 26 Testhomepages enthalten ein Foto der Zielperson. Von diesen 21 Fotos werden neun korrekt extrahiert, in elf Fällen wird das vorhandene Bild nicht extrahiert und in einem Fall wird eine andere Grafik anstatt des Fotos der Zielperson extrahiert. In den Fällen, in denen das vorhandene Bild nicht extrahiert wird, trifft die der Extraktionsmethode zu Grunde liegende Annahme, dass der Nachname der Zielperson im Dateinamen oder dem Alternativtext der Bilddatei enthalten ist, nicht zu. Vielmehr werden als Dateinamen die Initialen, der Vorname oder nur ein Teil des Nachnamens verwendet. Der Alternativtext wird überhaupt nur bei 7 der 21 vorhandenen Bilder genutzt. In einem Fall wird als Alternativtext der Begriff „Portrait“ verwendet, welcher nicht in der Liste der vorab gesammelten Begriffe enthalten ist.

Um diese Fehler zu vermeiden, müsste die Methode zur Extraktion der Bilddatei dahingehend erweitert werden, dass zum einen die Dateinamen auch nach Teilen des Namens oder Initialen der Zielperson durchsucht werden und zum anderen die Liste der in den Alternativtexten verwendeten Begriffe erweitert wird.

In 17 der 26 untersuchten Homepages sind die Publikationen der Zielperson oder eine Auswahl derer angegeben. In zehn Fällen werden die vorhandenen Publikationseinträge korrekt extrahiert, in den anderen Fällen sind die Ergebnisse fehlerhaft. Die Fehler und ihre Ursachen sind sehr unterschiedlich.

In zwei Fällen enthält die Menüleiste einen Link auf eine Seite mit den gesammelten Publikationen der Mitglieder des Fachbereichs, an dem die Zielperson tätig ist. Da dieser Link vor dem Link auf die Publikationsseite der Zielperson auftritt, wird er extrahiert und dementsprechend die Publikationen des gesamten Fachbereichs als die Publikationen der Zielperson dargestellt. In einer anderen Homepage wird das Label „Veröffentlichungen“ des Links auf die Publikationsseite als Textelement identifiziert, da in dem Link-Tag ein Font-Tag enthalten ist. Ein weiterer Fehler ist aufgetreten, weil der Link auf die Publikationsseiten in Form eines Bilds und nicht als Textlabel gesetzt und dementsprechend nicht erkannt wurde.

Eine andere Fehlerursache liegt darin, dass die Extraktion der Publikationseinträge aus der Unterseite kein Ergebnis liefert. Das Extraktionsverfahren geht von der Annahme aus, dass das Label des Links auf die Unterseite als Überschrift in der Unterseite wiederholt wird. Diese Annahme wird verwendet, um den Anfang des zu extrahierenden Bereichs zu bestimmen. In der betroffenen Homepage wird das Label aber nicht als Überschrift wiederholt und dementsprechend kein Eintrag extrahiert.

Eine weitere Homepage existiert nur auf Englisch, so dass die Stichwörter, die zum Lokalisieren der Publikationseinträge ermittelt wurden, zu keinem Treffer führen, da sie nur deutsche Begriffe abdecken.

Informationen zu Projekten der Zielperson sind nur in 8 der 26 Homepages angegeben. In vier dieser acht Fälle werden diese Einträge korrekt extrahiert, in den anderen werden die vorhandenen Einträge nicht oder falsch extrahiert. Die Ursachen für die Fehler sind in zwei Fällen dieselben, die bei der Extraktion der Publikationseinträge auftreten: Ein-

mal enthält die Homepage selbst einen Link zu den Projekten des Fachbereichs, von denen das Verfahren annimmt, dass es sich um die Projekte der Zielperson handelt. In dem anderen Fall wird der Verweis auf die Projektseite nicht als Link, sondern als Textelement identifiziert. In einem Fall sind die Angaben zu den Projekten der Zielperson auf der Hauptseite enthalten, aber ohne eine Überschrift, die diesen Textteil explizit benennt. In dem vierten Fall, in dem die Projektangaben nicht extrahiert werden, obwohl sie vorhanden sind, kann eine Ursache nicht eindeutig benannt werden.

Der Lebenslauf der Zielperson ist lediglich in 8 der 26 Homepages angegeben. Das Extraktionsverfahren liefert aber nur in einem Fall das korrekte Ergebnis, in den anderen Fällen findet es den Lebenslauf nicht. Für das Nicht-Auffinden der vorhandenen Information gibt es zwei Gründe: In fünf Fällen wird die Person am Anfang des Inhaltsbereiches der Homepage kurz vorgestellt, ohne dass dieser Teil mit einer Überschrift explizit als Lebenslauf bezeichnet wird. Daher werden die vorhandenen Informationen von dem Verfahren nicht gefunden. In den drei anderen Fällen befindet sich der Lebenslauf auf einer Unterseite, wobei die für die Label der Links auf diese Unterseiten verwendeten Begriffe „Kurzvita“, „Person“ und „Portrait“ sind. Diese Begriffe werden durch die vorab erstellte Liste der zur Bezeichnung des Lebenslaufs verwendeten Schlüsselwörter nicht abgedeckt und die Links werden dementsprechend nicht identifiziert. Diese Fehler können durch eine entsprechende Erweiterung der Liste der Schlüsselwörter leicht beseitigt werden. Um die erst genannten Fehler zu vermeiden, müsste ein Verfahren entwickelt werden, dass sich charakteristische Eigenschaften von Lebensläufen wie z.B. den tabellarischen Aufbau zu Nutze macht, um diese zu erkennen und zu extrahieren.

Im folgenden Kapitel werden die Ergebnisse dieser Evaluierung zusammengefasst und daraus resultierende Verbesserungsmöglichkeiten aufgezeigt.

5.3 Zusammenfassung der Ergebnisse und Verbesserungsmöglichkeiten

Die Extraktion der Publikationseinträge aus der DBLP und CiteSeer liefert in den meisten Fällen das richtige Ergebnis, jedoch hat sich gezeigt, dass nur zu einem geringen Teil der Testpersonen Einträge in CiteSeer vorhanden sind und auch die DBLP in vielen Fällen lediglich eine Veröffentlichung aufführt, nämlich die Veröffentlichung in dem ISI 2004-Konferenzband. Darüber hinaus ist der CiteSeer-Server nicht stabil und oft kann

die Verbindung für das Stellen einer Anfrage nicht hergestellt werden. Diese Gründe sprechen zusätzlich zu den in Kapitel 3.2.1.2 erläuterten für die Einbindung der Digitalen Bibliothek DAFFODIL. Zum einen greift DAFFODIL auf mehr Quellsysteme zu, so dass es nicht so stark von der Verfügbarkeit eines Systems abhängig ist und mehr Ergebnisse liefern kann als die Suche in CiteSeer und der DBLP allein. Zum anderen ist in DAFFODIL bereits die Fusion der Extraktionsergebnisse aus den einzelnen Quellsystemen realisiert. So kann die Effektivität und Effizienz der Suche nach den Publikationen der Zielpersonen verbessert werden. Die Suche mit DAFFODIL kann ähnlich zu der Suche mit Hilfe der Google-API mit einem SOAP-Client umgesetzt werden. Die dazu benötigte Java-Klasse wird in dem Wiki[35] der Entwicklergruppe von DAFFODIL zur Verfügung gestellt.

Die spezialisierte Suche nach Homepages liefert mit einem Recall von 0.65 und einer Precision von 0.71 eine gute Ausgangsbasis für Optimierungsansätze. Die Fälle, in denen die Homepages nicht gefunden wurden, obwohl sie in den Ergebnissen der Google-Suche enthalten waren, sind auf verschiedene Fehlerquellen zurückzuführen, so dass nicht auf eine eindeutige Verbesserungsmöglichkeit geschlossen werden kann. Es bietet sich hingegen eine Kombination verschiedener Optimierungsansätze an, deren Potenzial in einer Evaluierungsstudie zu untersuchen ist. Diese sind die Überarbeitung der Bewertungsfunktion und der verwendeten Heuristiken, die Verwendung eines Ranking-Verfahrens sowie die Einführung eines Ansatzes zur Anfrageerweiterung.

Der erste Schritt zur Verbesserung des Verfahrens ist folglich die Überarbeitung der Bewertungsfunktion. Es ist experimentell zu untersuchen, ob wirklich alle der ermittelten Eigenschaften für die Bestimmung der Homepage eines Informationswissenschaftlers gleich relevant sind oder ob eine Gewichtung der verschiedenen Eigenschaften dazu führt, dass mehr Homepages gefunden und weniger falsche Ergebnisse geliefert werden. Ebenso ist an einer genügend großen Testmenge zu prüfen, ob weitere Eigenschaften in die Bewertung miteinbezogen werden können.

Darüber hinaus wird angenommen, dass der Recall des Suchverfahrens durch die Umstellung auf eine Ranking-Lösung verbessert werden kann: Als Ergebnis der Suche wird eine nach Relevanz sortierte Liste der Ergebnisse der Google-Suche erstellt, anstatt das Ergebnis auf die am höchsten bewertete Seite einzuschränken. Die Rangfolge der Ergebnisseiten würde auf dem im Laufe des Bewertungsverfahrens erstellten Ranking

[35] http://www.is.informatik.uni-duisburg.de/wiki/index.php/SOAP_Gateway (verifiziert am 29.03.2006)

beruhen, dementsprechend ist dann die Wahrscheinlichkeit, dass es sich bei einer Ergebnisseite um die Homepage der Zielperson handelt, um so größer, desto besser die Position der Seite in der gerankten Liste der Ergebnisse ist. Aus Effizienzgründen scheint es nicht sinnvoll, die Methoden zur Informationsextraktion auf alle Ergebnisseiten anzuwenden. Vielmehr müsste ein Schwellenwert für die Bewertungspunktzahl verwendet werden, unter welchem es sich bei einer Seite mit großer Wahrscheinlichkeit nicht um die Homepage der Zielperson handelt. Folglich würden die Extraktionsmethoden nur auf die Seiten angewandt werden, deren Bewertung über diesem Schwellenwert liegt. Als Schwellenwert könnte derselbe Wert verwendet werden, der in der Evaluierung der Bewertungsfunktion ermittelt wurde, wobei auch dieser noch an einer größeren Testmenge evaluiert werden müsste. Als Ergebnis würde dem Nutzer schließlich eine nach Relevanz sortierte Liste der Homepage-Kandidaten mit den aus der jeweiligen Seite extrahierten Informationen präsentiert.

Es wird angenommen, dass durch diese Erweiterung die Anzahl gefundener Homepages zunimmt und sich so der Recall des Suchverfahrens erhöht, gleichzeitig aber die Precision abnimmt, da in der Regel nur eine der Ergebnisseiten die Homepage der Zielperson ist und die anderen Ergebnisseiten nicht relevant sind. Bei dieser Lösung kann es in Hinblick auf die Nutzerzufriedenheit problematisch werden, dass die Suche nach den Homepages und die Informationsextraktion aus diesen zur Anfragezeit durchgeführt werden. Denn es ist davon auszugehen, dass die Suchzeit stark zunimmt, wenn die Extraktionsmethoden auf mehrere Homepages angewandt werden.

Eine zusätzliche Möglichkeit zur Verbesserung der Suchergebnisse kann zudem die Umsetzung einer „flexiblen“ Expansion der an Google gestellten Anfrage sein. In 3.2.3.1 ist beschrieben worden, dass die Anfrageerweiterung mit vorab festgelegten Termen wie „Homepage“ oder „Informationswissenschaft“ nicht zu einer durchgängigen Verbesserung der Positionen der Homepages im Ranking der Google-Ergebnisse geführt hat. Alternativ könnte daher die Anfrage mit für die jeweilige Zielperson spezifischen Begriffen erweitert werden, die aus der Menge der Schlagwörter, welche in den Titeln der Publikationen der Zielperson am häufigsten vorkommen, stammen. Wenn so die Positionen der Homepages in der gerankten Liste der Google-Ergebnisse verbessert werden, kann in Verbindung mit einer höheren Gewichtung der Google-Rankingposition in der Bewertungsfunktion vielleicht eine höhere Precision für das Suchverfahren erreicht werden.

Zudem ist anzunehmen, dass durch diese Art der Anfrageerweiterung das Problem der Namensambiguität an Bedeutung verliert, da die Wahrscheinlichkeit abnimmt, dass Seiten von eventuell vorhandenen Namensvettern der Zielperson unter den Ergebnissen sind. So werden voraussichtlich weniger andere Seiten fälschlicherweise als Homepage identifiziert und die Precision nimmt zu. Darüber hinaus wird davon ausgegangen, dass mit der erweiterten Anfrage die Anzahl der Fälle abnimmt, in denen die Homepage nicht in den ersten zehn Google-Ergebnissen enthalten ist und gar nicht in dem Bewertungsverfahren berücksichtigt wird, so dass sich die Menge gefundener Homepages und somit der Recall vergrößert.

Die beschriebene Methode kann mit Hilfe einer von DAFFODIL bereitgestellten Funktion, die auch über den Web Service eingebunden werden kann, leicht realisiert werden. Diese extrahiert die in den Titeln der Publikationen vorkommenden Schlüsselwörter und erstellt aus ihnen eine nach der Häufigkeit des Auftretens sortierte Liste.

Die Extraktion der Informationen aus den Homepages liefert gute Ergebnisse, die Annahmen zu dem Aufbau von persönlichen Homepages und der Platzierung der Informationsobjekte haben sich damit weitestgehend als richtig erwiesen. Die aufgetretenen Fehler sind darauf zurückzuführen, dass die entwickelten Heuristiken zu den verwendeten Schlüsselwörtern nicht ausreichend sind. So werden vorhandene Informationen nicht gefunden, weil die in den Homepages verwendeten Schlüsselwörter zur Kennzeichnung der zu extrahierenden Bereiche nicht mit den vorab ermittelten Begriffen übereinstimmen. Dies ist der Fall bei der Extraktion der akademischen Titel, der Fotos und dem Lebenslauf. Ähnlich dazu werden viele Bilder nicht identifiziert, weil die Annahmen zur die Benennung der Bilddateien und der in den Alternativtexten verwendeten Begriffe nicht zutreffen. Um den Recall der Extraktionsverfahren zu verbessern, müssen folglich die Listen der akademischen Titel und der für die Bezeichnung des Lebenslaufs verwendeten Schlüsselwörter erweitert werden. Analog kann der Recall der Extraktion der Bilddateien verbessert werden, indem das Verfahren zur Identifikation der Bilddatei um zusätzliche Möglichkeiten der Dateibenennung erweitert und die Liste der in den Alternativtexten verwendeten Begriffe entsprechend angepasst wird.

Damit diese Erweiterungen möglichst umfassend sind, sollte eine größere Menge von Homepages untersucht werden, denn es ist anzunehmen, dass auch die Vereinigungs-

menge der in der Trainingsmenge und der Testmenge enthaltenen Schlüsselwörter und Eigenschaften nicht ausreichend ist.

Bei der Analyse der Extraktionsergebnisse der zusammenhängenden Informationen Publikationen, Projekte und Lebenslauf ist aufgefallen, dass die Annahme, das Label des Links auf die Unterseite werde als Überschrift in der Unterseite wiederholt, ungenügend ist, um den zu extrahierenden Bereich zu bestimmen. Darüber hinaus führt die Problematik, dass Anfang und Ende der zu extrahierenden Bereiche nicht genau bestimmt werden, dazu, dass neben der tatsächlichen Zielinformation viel Text extrahiert wird, der für den Nutzer irrelevant ist. Eine Möglichkeit zur Vermeidung dieser Fehler könnte die Beschränkung der Anwendung der Extraktionsmethoden auf den inhaltstragenden Bereich der jeweiligen Webseite sein. Das heißt, dass die nicht-inhaltstragenden Bereiche wie Navigationsleiste, Kopf- und Fußzeile von der Anwendung der Informationsextraktionsmethoden ausgeschlossen werden. Das Erkennen dieser verschiedenen Teilbereiche einer Webseite ist das Ziel der Verfahren zur Segmentierung von Webseiten, wie in 2.3.2.1 beschrieben wurde. Zwei solcher Verfahren sollen an dieser Stelle kurz vorgestellt werden.

Kovačevic et al. [2002] haben ein Verfahren zur Erkennung häufig auftretender Bereiche in Webseiten entwickelt. Diese Bereiche sind „header, footer, left menu, right menu, center of the page“ [Kovačevic et al. 2002, 4]. Zur Identifizierung dieser Bereiche haben sie Heuristiken erstellt, die auf der Position der einzelnen Elemente in einem Strukturbaum und der Positionierung in einem Browserfenster aufbauen.

Lewandowski [2005b, 217ff.] schlägt aufbauend auf dem Ansatz von Wang und Hu [2002] zur Erkennung von echten Tabellen in Webseiten ein anderes Verfahren vor:

> „Bei einem Aufbau des Dokuments mit Hilfe von Tabellen muss diejenige Tabellenspalte bzw. -zelle gefunden werden, in der die tatsächlichen Inhalte stehen. Ist das Dokument ohne Tabellen aufgebaut, so müssen die verschiedenen Dokumente eines Servers [in diesem Fall die Haupt- und Unterseiten der jeweiligen Homepage, Anm. d. Verf.] miteinander verglichen werden, um gleichlautende Elemente entfernen zu können“
>
> [Lewandowski 2005b, 217]

Für den ersten Fall schlägt er die Verwendung des von Wang und Hu [2002] entwickelten Verfahrens zur Identifikation von Tabellen in Webseiten vor. Dieser mit Methoden des Maschinellen Lernens realisierte Ansatz hat zum Ziel, Tabellen, die der Präsentation zusammenhängender Daten dienen, von den Tabellen zu unterscheiden, die zu reinen Layout-Zwecken verwendet werden. Laut Lewandowski kann dieses Verfahren auch

umgekehrt dazu angewandt werden, „den inhaltstragenden Teil von aus Gründen des Layouts verwendeten Tabellen zu extrahieren“ [Lewandowski 2005b, 220].

Der Einsatz eines derartigen Verfahrens zur Segmentierung der Homepages und der Konzentration der Anwendung der Extraktionsregeln auf den inhaltstragenden Bereich der jeweiligen Seite dürfte nicht nur die Ergebnisse der Extraktion der zusammenhängenden Informationen verbessern, sondern auch die der anderen Extraktionsmethoden. Denn in den Fällen, in denen falsche Informationsobjekte extrahiert wurden, wie z.B. eine Grafik, die der Gestaltung der Webseite dient, anstatt des Fotos der Zielperson, befanden sich diese häufig in den nicht-inhaltstragenden Bereichen der entsprechenden Homepage.

6 Zusammenfassung und Erweiterungsmöglichkeiten

Ziel dieser Arbeit war, ein Verfahren zur Suche nach Informationen zu wissenschaftlich tätigen Personen prototypisch für den Bereich der Informationswissenschaften im deutschsprachigen Raum zu entwickeln, um verwendbare Techniken und Quellen zu identifizieren und die generelle Umsetzbarkeit eines solchen Verfahren zu untersuchen.

So ist ein System entworfen und in einer Java-Applikation implementiert worden, das ausgehend vom Namen eines Wissenschaftlers nach den Informationsitems akademischer Titel, E-Mail-Adresse, Foto der Zielperson, Publikationen, Projekte und Lebenslauf sucht. Dazu wurden als Quellen für die Veröffentlichungsdaten die Publikationsdienste DBLP und CiteSeer sowie für die weiteren Informationsobjekte die persönliche Homepage der jeweiligen Zielperson ausgewählt. Aus den Ergebnisseiten der Quellsysteme und den Homepages werden die Informationen, die von Interesse sind, zur Anfragezeit extrahiert und dem Suchenden in der Benutzeroberfläche in integrierter Form präsentiert.

Für die Erschließung der DBLP und von CiteSeer sind manuell Wrapper implementiert worden, die Layout und Strukturierung der Ergebnisseiten nutzen, um die Publikationseinträge zu extrahieren. Die extrahierten Daten werden auf Basis der Veröffentlichungstitel verglichen und in einer Liste unter Angabe des Fundortes fusioniert dargestellt.

Für das Auffinden der persönlichen Homepage der jeweiligen Zielperson ist ein spezialisiertes Suchverfahren entwickelt worden, das mit Hilfe einer Bewertungsfunktion die Homepage aus den Ergebnisseiten einer Google-Suche mit dem Namen als Anfrageterm herausfiltert. Die Bewertungsfunktion beruht auf typischen Eigenschaften von persönlichen Homepages von Informationswissenschaftlern, die in der Analyse einer Trainingsmenge ermittelt worden sind.

Zur Extraktion der Informationsitems aus den Homepages sind Heuristiken zu Aufbau und Struktur der Homepages und der Verwendung von Schlüsselwörtern zur Bezeichnung bestimmter Inhaltsbereiche und Unterseiten erstellt worden. Die Methoden zur Informationsextraktion basieren auf diesen Heuristiken, die sich zunutzemachen, dass die Homepages bestimmten Konventionen unterliegen und somit eine relativ homogene Menge darstellen.

Die Evaluierung des Verfahrens in Form eines Vergleichs mit den Ergebnissen einer manuellen Suche nach den Zielinformationen zu einer Gruppe von Testpersonen hat

gezeigt, dass das Verfahren gute Ergebnisse liefert und der gewählte Ansatz insgesamt Erfolg versprechend ist. Das spezialisierte Suchverfahren zur Homepage-Suche erzielt in der Evaluierung einen Recall von 0.65 und eine Precision von 0.71. Es wird erwartet, dass durch eine Überarbeitung der verwendeten Bewertungsfunktion und eine Gewichtung der in dieser berücksichtigten Eigenschaften sowie die Einführung eines Ranking-Verfahrens die Suchergebnisse verbessert werden können.

Hinsichtlich der Informationsextraktion aus den Homepages hat sich gezeigt, dass die Homepages der Informationswissenschaftler genügend homogen sind, um Heuristiken zu Struktur und Aufbau der Seiten in den Extraktionsmethoden einzusetzen. Eine Erweiterung der Liste der zur Bezeichnung bestimmter Inhalte verwendeten Begriffe, die anhand einer Untersuchung einer größeren Trainingsmenge ermittelt werden können, kann den Recall der Extraktionsmethoden verbessern. Darüber hinaus wird angenommen, dass der Einsatz eines Verfahrens zur Segmentierung von Webseiten die ausschließliche Anwendung der Extraktionsregeln auf den inhaltstragenden Bereich der Seiten ermöglicht und so weniger Fehler in der Extraktion auftreten.

Neben diesen Ansätzen zur direkten Verbesserung des entwickelten Verfahrens ergeben sich in Hinblick auf die verwendeten Quellen und Suchmaschinen sowie die generelle Funktionalität des Systems einige Erweiterungsmöglichkeiten.

Bezüglich der Quellenauswahl hat sich gezeigt, dass die Homepages der Zielpersonen in vielen Fällen nicht ausreichen, um die Zielinformationen zu ermitteln. Zum einen verfügen nicht alle wissenschaftlich tätigen Personen über eine persönliche Homepage, wie während der Erstellung der Testmenge deutlich wurde. Vor allem Personen, die nicht an öffentlichen Forschungseinrichtungen wie z.B. Universitäten sondern in der Wirtschaft tätig sind, besitzen in der Regel keine persönliche Homepage. Darüber hinaus hat sich herausgestellt, dass selbst wenn eine Zielperson eine Homepage hat, oft nicht alle Zielinformationen in dieser angegeben sind. Gerade Informationen zu Projekten und Lebenslauf sind häufig nicht vorhanden. Eine Erweiterung der Quellenauswahl erscheint also sinnvoll. Es könnte untersucht werden, ob andere Seiten der Google-Ergebnisse relevante Informationen enthalten und wie diese extrahiert werden können. Darüber hinaus wäre zu prüfen, ob es Webangebote gibt, die Informationen zu Informationswissenschaftlern gruppieren. So ist z.B. während der Evaluierung die Seite *www.competence-site.de* aufgefallen, die eine Art Expertennetzwerk für Wissenschaftler und Praktiker für die Bereiche Management und IT ist und auch Themen der Informati-

onswissenschaft abdeckt. Zu jedem der eingetragenen Experten werden der Lebenslauf, die Organisation, die Kernkompetenzen und ein Foto bereitgestellt.

Neben der Einbeziehung zusätzlicher Quellen ist auch zu überlegen, weitere Suchmaschinen zum Auffinden der Homepages hinzuzuziehen. Der Prototyp verwendet Google als Web Service, wobei die Suchfunktion über die Einbindung der Methoden der Google API realisiert ist. Während der Evaluierung hat sich gezeigt, dass der Web Service relativ instabil ist und oft nicht zur Verfügung steht. Darüber hinaus sind die Ergebnisse der Suche oft nicht in der gleichen Qualität wie die Ergebnisse der Onlinesuche mit Google. So werden z.B. manche Seiten, die in der Ergebnisliste der Onlinesuche an erster Position aufgeführt werden, in der Ergebnisliste der Suche mit der API gar nicht aufgelistet. Es konnte zu diesem Problem keine offizielle Stellungnahme von Google gefunden werden, es ist aber anzunehmen, dass die Unterschiede in den Ergebnissen auf die Verwendung verschiedener Indexe zurückzuführen sind, die nicht auf dem gleichen Stand sind. Die Verwendung anderer Suchmaschinen oder einer Metasuchmaschine, bzw. die Entwicklung eines eigenen Metasuchverfahrens, das mehrere Suchmaschinen einbezieht, erscheint daher sinnvoll. Interessant wäre auch, zu untersuchen, ob eine spezielle Suchmaschine sich besonders gut für die Suche nach persönlichen Homepages von Wissenschaftlern eignet.

Das umgesetzte Verfahren wurde prototypisch für den Bereich der Informationswissenschaften entwickelt, aber von Interesse ist natürlich auch die Übertragbarkeit auf andere Wissenschaftsbereiche. Folgende Änderungen bzw. Untersuchungen müssten dazu vorgenommen werden: Zunächst müssen adäquate Quellen für Publikationsangaben für den ausgewählten Bereich ermittelt und Wrapper für diese Quellsysteme erstellt werden. Bezüglich der speziellen Suche nach den Homepages müssen die in der Bewertungsfunktion verwendeten typischen Eigenschaften, die spezifisch für den Bereich der Informationswissenschaften sind, an den entsprechenden Bereich angepasst werden. Ein Beispiel ist das Vorkommen von Begriffen wie „Informationswissenschaft“ oder „Informatik“ im Titel oder Snippet der Google-Ergebnisse. Generell wäre zu überprüfen, ob sich die persönlichen Homepages überhaupt als Quelle für den jeweiligen Wissenschaftsbereich eignen, d. h., ob die diesem Bereich angehörigen Wissenschaftler in der

Regel eine persönliche Homepage besitzen und darüber hinaus ähnliche Konventionen bezüglich des Aufbaus und der verwendeten Begriffe vorherrschen.

Die Aufhebung der Einschränkung auf einen Wissenschaftsbereich erscheint nur begrenzt sinnvoll. Es ist anzunehmen, dass mit einer Vergrößerung der Menge der potenziellen Zielpersonen die Wahrscheinlichkeit von Namensambiguitäten zunimmt und problematisch wird. Um dieses Problem zu umgehen, ist folgende Lösung denkbar: Vorab werden für eine begrenzte Menge von Wissenschaftsbereichen Wrapper für adäquate Publikationsdienste und einzelne Bewertungsverfahren für das Herausfiltern der Homepages aus den Google-Ergebnissen erstellt. Der Nutzer wählt zusätzlich zu der Eingabe des Namens der Zielperson den Wissenschaftsbereich aus, in den er diese einordnet, und das System verwendet die für den ausgewählten Bereich vorgesehenen Quellen sowie die entsprechende Bewertungsfunktion. Alternativ dazu könnte ein Verfahren verwendet werden, dass mehr als zehn Google-Ergebnisse berücksichtigt und diese so gruppiert, dass die Seiten, welche sich auf ein und dieselbe Person beziehen, in einer Gruppe zusammengefasst sind. Das System könnte dann in den einzelnen Gruppen nach der Homepage der zugehörigen Person suchen und von den in den Ergebnisseiten vorkommenden Begriffen auf den Wissenschaftsbereich schließen sowie die relevanten Publikationsdienste auswählen. Dem Nutzer wird dann gegebenenfalls eine Liste von Personen mit den zu ihnen gefundenen Informationen präsentiert. Al-Kamha und Embley [2004] stellen ein solches Verfahren zur Gruppierung von „search-engine returned citations for person-name queries“ vor [Al-Kamha, Embley 2004, 96].

Eine andere Möglichkeit der Erweiterung ist, das Verfahren auf andere Sprachräume auszuweiten und somit multilingual zu gestalten. Gerade vor dem Hintergrund, dass viele Wissenschaftler nur eine englische Version ihrer Homepage zur Verfügung stellen, erscheint dies sinnvoll. In einem multilingualen System können die Quellen für die Publikationsdaten weiter verwendet werden, da sie Publikationen aus internationalen Veröffentlichungen enthalten. Bezüglich der Verwendung von Homepages als Quelle müsste untersucht werden, ob die ermittelten typischen Eigenschaften hinsichtlich der Struktur und der verwendeten Begriffe auch auf Homepages in anderen Sprach- bzw. Kulturräumen zutreffen. Ist dies der Fall, müssen die in der Bewertungsfunktion und den Extraktionsmethoden verwendeten Begriffe wie „Veröffentlichungen“ oder „Lebenslauf“ in die entsprechenden Sprachen übersetzt werden. Zusätzlich wäre zu überlegen, ob der

Nutzer mit seiner Anfrage den Sprachraum der Zielperson angibt oder ob das System ohne Spracheinschränkung nach Homepages sucht. In ersterem Fall grenzt dann das System die Suche auf Dokumente der ausgewählten Sprache ein und verwendet jeweils die sprachspezifische Bewertungsfunktion und Extraktionsmethode. In letzterem Fall muss das System mit Hilfe eines Spracherkennungsverfahren die Sprache der gefundenen Homepage identifizieren, um die entsprechende Extraktionsmethode einzusetzen.

Bevor jedoch die Umsetzung dieser Erweiterungsmöglichkeiten und die Einbindung neuer Quellen und anderer Suchmaschinen untersucht wird, sollten die in der Analyse dargestellten Verbesserungsvorschläge für das bestehende Verfahren umgesetzt werden. Zusammengefasst ergeben sich so folgende Aufgaben: Die virtuelle Digitale Bibliothek DAFFODIL sollte als Quelle für die Publikationsdaten eingebunden werden, so dass die Nutzung der DBLP und CiteSeer wegfallen. Anhand einer umfassenden Testmenge sollten die in der Informationsextraktion aus den Homepages verwendeten Heuristiken erweitert und verfeinert werden. Die Trainingsmenge kann auch dazu verwendet werden, die in dem spezialisierten Suchverfahren eingesetzte Bewertungsfunktion genauer zu evaluieren. Darüber hinaus sollte das Suchverfahren um das beschriebene Rankingverfahren erweitert werden.

Literaturverzeichnis

[Al-Kamha, Embley 2004]

Al-Kamha, Reema, Embley, David: Grouping Search-Engine returned Citations for Person-Name Queries. In: Laender, Alberto H.F.; Lee, Dongown; Ronthaler, Marc (Hrsg.) (2004): Sixth ACM CIKM International Workshop on Web Information and Data Management (WIDM 2004). ACM Press, S. 96-103.

[Appelt, Israel 1999]

Appelt, D.; Israel, David (1999): Introduction to Information Extraction Technology.
A Tutorial Prepared for IJCAI-99, SRI International.
http://www.ai.sri.com/~appelt/ie-tutorial/IJCAI99.pdf (verifiziert am 28.03.2006)

[Berendt et al. 2002]

Berendt, Bettina; Hotho, Andreas. Stumme, Gerd (2002): Towards Semantic Web Mining. In: Horrocks, Ian; Hendler, James (Hrsg.) (2002): The Semantic Web - International Semantic Web Conference 2002. Berlin et al.: Springer, S. 264-278. [Lecture Notes in Computer Science 2342]

[Bergman 2001]

Bergman, Michael (2001): The DeepWeb: Surfacing Hidden Value.
http://beta.brightplanet.com/deepcontent/tutorials/DeepWeb/index.asp
(verifiziert am 28.03.2006)

[Bilenko et al. 2003]

Bilenko, Mikhail; Mooney, Raymond J.; Cohen, William W.; Ravikumar, Pradeep; Fienberg, Stephen (2003): Adaptive Name Matching in Information Integration. In: Intelligent Systems Vol. 18(5), S. 16-23.

[Brin, Page 1998]

Brin, S.; Page, L. (1998): The anatomy of a large-scale hypertextual web search engine. In: Computer Networks Vol. 30(1-7), S. 107-117.

[Chakrabarti 2003]
Chakrabarti, Soumen (2003): Mining the Web: Discovering Knowledge from Hypertext Data. San Francisco: Morgen Kaufmann.

[Chakrabarti et al. 1999]
Chakrabarti, Soumen; van den Berg, Martin; Dom, Byron (1999): Focused Crawling: a new approach to topic-specific Web resource discovery.
In: Computer Networks Vol. 31(11-16), S. 1623-1640.

[Chau et al. 2003]
Chau, Michael; Zeng, Daniel; Chen, Hsinchun; Huang, Michael; Hendriawan, David (2003): Design and evaluation of a multi-agent collaborative Web mining system.
In: Decision Support Systems Vol. 35(1), S. 167-183.

[Chen, Chue 2005]
Chen, Lihui; Chue, Wai Lian (2005): Using Web structure and summarisation techniques for Web content mining. In: Information Processing and Management
Vol. 41(5), S. 1225-1242.

[Cohen et al. 2000]
Cohen, William W.; McCallum, Andrew; Quass, Dallan (2000): Learning to Understand the Web. In: IEEE Data Engineering Bulletin 23(3), S. 17-24.

[Cooley et al. 1997]
Cooley, Robert; Mobasher, Bamshad; Srivastava, Jaideep (1997): Web Mining: Information and Pattern Discovery on the World Wide Web. In: Proceedings of the
9th International Conference on Tools with Artificial Intelligence (ICTAI 1997),
S. 558-567.

[Eikvil 1999]
Eikvil, Line (1999): Information Extraction from the World Wide Web - A survey.
http://www.nr.no/documents/samba/research_areas/BAMG/Publications/webIE_rep945.ps
(verifiziert am 28.03.2006)

[Etzioni 1996]

Etzioni, Oren (1996): The World-Wide Web: Quagmire or Gold Mine? In: Communications of the ACM 39(11), 1996, S. 65-68.

[Ferber 2003]

Ferber, Reginald (2003): Information Retrieval - Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web. Heidelberg: dpunkt.verlag.

[Fuhr et al. 2000]

Fuhr, Norbert; Gövert, Norbert; Klas, Claus-Peter: An Agent-Based Architecture for Supporting High-Level Search Activities in Federated Digital Libraries. In: Proceedings of the 3rd International Conference of Asian Digital Library, 2000, S. 247-254.

[Garofalakis et al. 1999]

Garofalakis, M. N.; Rastogi, R.; Seshadri, S.; Shim, K. (1999): Data mining and the Web: past, present and future. In: C. Shahabi (Hrsg.) (1999): Proceedings of the 2nd international Workshop on Web information and Data Management. WIDM '99.
New York: ACM Press, S. 43-47.

[Goodrum et al. 2001]

Goodrum, Abby A.; McCain, Katherine W.; Lawrence, Steve; Giles, Lee G. (2001): Scholarly publishing in the Internet age: a citation analysis of computer science literature. In: Information Processing and Management Vol.37. S. 661-675.

[Google 2006]

About Google Scholar.
http://scholar.google.com/scholar/about.html
(verifiziert am 28.03.2006)

[Han et al. 2004]

Han, Hui; Giles, C. Lee; Zha, Hongyuan; Li, Cheng; Tsioutsiouliklis, Kostas (2004): Two supervised learning approaches for name disambiguation in author citations.
In: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries 2004,
S. 296-305.

[Hawking, Craswell 2002]

Hawking, David; Craswell, Nick (2002): Overview of the TREC-2001 Web Track. In: Voorhees, E.M.; Harman, D.K. (Hrsg.) (2002): The Tenth Text Retrieval Conference (TREC 2001). NIST, S. 61-67.

[Hoff 2002]

Hoff, Gerd (2002): Ein Verfahren zur thematisch spezialisierten Suche im Web und seine Realisierung im Prototypen HomePageSearch.
Fachbereich IV der Universität Trier: 2002.

[Hoff, Mundhenk 2001]

Hoff, Gerd; Mundhenk, Martin (2001): Creating a virtual library with HPSearch and Mops. Vortrag auf der IuK Conference - Information and Communication of the Learned Societies in Germany 2001.
http://www.informatik.uni-trier.de/~mundhenk/virt-lib/ (verifiziert am 28.03.2006)

[Hsu et al. 1998]

Hsu, Chun-Nan; Dung, Ming-Tzung (1998): Generating Finite-state Transducers for semi-structured Data Extraction from the web. In: Information systems Vol. 23(8).
S. 521-538.

[Kambhampati, Knoblock 2003]

Kambhampati, Subbarao; Knoblock, Craig A. (2003): Information Integration on the Web. In: Intelligent Systems Vol. 18(5), S. 14-15.

[Klas et al. 2005]

Klas, Claus-Peter; Kriewel, Sascha; Fuhr, Norbert; Schaefer André (2005): DAFFODIL - Nutzerorientiertes Zugangssystem für heterogene Digitale Bibliotheken. In: Okenfeld, Marlies (Hrsg.): Leitbild Informationskompetenz Positionen - Praxis - Perspektiven im europäischen Wissensmarkt. 27. Online-Tagung der DGI. Frankfurt a.M.: DGI.

[Klopotek 2003]

Klopotek, Miczyslaw (2003): Intelligent Information Retrieval on the Web. In: Szczepaniak et al. (Hrsg.) (2003): Intelligent Exploration of the Web.
Heidelberg: Physica-Verlag, S. 57-73.

[Kosala, Blockeel 2000]

Kosala, Raymond; Blockeel, Hendrik (2000): Web Mining Research: A Survey.
In: SIGKDD Explorations Vol. 2(1), S. 1-15.

[Kovačevic et al. 2002]

Kovačevic, Milos; Diligenti, Michelangelo; Gori, Marco; Milutinovic, Veljko M. (2002): Recognition of Common Areas in a Web Page Using Visual Information: a possible application in a page classification. In: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002). IEEE Computer Society Press, S. 250-257.

[Laender et al. 2002]

Laender, Alberto; Ribeiro-Neto, Berthier; da Silva, Altigtan; Teixeira, Juliana: A Brief Survey of Web Data Extraction Tools. In: SIGMOD Record, Vol. 31(2), 2002, S. 84-93.

[Lawrence et al. 1999]

Lawrence, Steve; Giles, C. Lee; Bollacker, Kurt: Digital Libraries and Autonomous Citation Indexing. In: IEEE Computer, Vol. 32(6), 1999, S. 67-71.

[Lewandowski 2004]

Lewandowski, Dirk (2004): Spezialsuche für wissenschaftliche Informationen. http://www.durchdenken.de/lewandowski/doc/suchmaschinen-news_dez2004.pdf (verifiziert am 28.03.2006)

[Lewandowski 2005a]

Lewandwoski, Dirk (2005a): Google Scholar - Aufbau und strategische Ausrichtung des Angebots sowie Auswirkung auf andere Angebote im Bereich der wissenschaftlichen Suchmaschinen. Gutachten im Auftrag des Hochschulbibliothekszentrums NRW, Februar 2005.
http://www.durchdenken.de/lewandowski/doc/Expertise_Google-Scholar.pdf
(verifiziert am 28.03.2006)

[Lewandowski 2005b]

Dirk Lewandowski (2005b): Web Information Retrieval - Technologien zur Informationssuche im Internet. DGI-Schrift (Informationswissenschaft 7).

[Ley 1997]

Ley, Michael (1997): Die Trierer Informatik-Bibliographie DBLP. In: Matthias Jarke, Matthias; Pasedach, Klaus; Pohl, Klaus (Hrsg.) (1997): Informatik '97, Informatik als Innovationsmotor, 27. Jahrestagung der Gesellschaft für Informatik, S. 257-266

[Ley 2002]

Ley, Michael (2002): The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives. In: Laender, Alberto; Oliveira, Arlindo (Hrsg.) (2002): Proceedings of 9th international symposium on String processing and information retrieval (SPIRE 2002). Berlin et al.: Springer, S. 1-10.

[Liu, Chang 2004]

Liu, Bing; Chang, Kevon Chen-Chuan (2004): Editorial. Special Issue on Web Content Mining. In: SIGKDD Explorations Vol. 6(2), S. 1-4.

[Luton 2002]

Luton, Tony (2002): Web Content Mining with Java. Chichester, England: John Wiley.

[Madria et al. 1999]

Madria, Sanjay; Bhomwick, Souray; Ng, Wee Keong; Lim, Ee-Peng (1999): Research Issues in Web Data Mining. In: Mohania, Mukesh; Min Tjoa, A. (Hrsg.) (1999): Proceedings of the First International Conference on Data Warehousing and Knowledge Discovery, DaWAK 1999. Berlin et al.: Springer, S. 303-312.
[Lecture Notes in Computer Science 1676]

[Mayer, Walter 2005]

Mayr, Philip; Walter, Anne-Kathrin (2005): Google Scholar - Wie tief gräbt diese Suchmaschine. Folien zum Vortrag auf der 11. IuK-Jahrestagung 2005.
http://www.ib.hu-berlin.de/~mayr/arbeiten/mayr_walter_iuk05.pdf
(verifiziert am 28.03.2006)

[Mueller 2004]

Mueller, John Paul (2004): GOOGLE Web Services - Building Applications with the GOOGLE API. San Francisco: Sybex.

[Muslea 1999]

Muslea, Ion (1999): Extraction Patterns for Information Extraction Tasks: A Survey. In: AAAI 1999 Workshop on Machine Learning for Information Extraction.

[Neumann 2001]

Neumann, Günter (2001): Informationsextraktion. In Klabunde et al. (Hrsg.) (2001): Computerlinguistik und Sprachtechnologie - Eine Einführung. Heidelberg: Spektrum Akademischer Verlag, 2001.

[Petinot et al. 2004]

Petinot, Yves; Giles, C. Lee; Bhatnagar, Vivek; Teregowda, Pradeep B.; Han, Hui; Councill, Isaac (2004): CiteSeer-API: Towards Seamless Resource Location and Interlinking for Digital Libraries. In: Proceedings of the ACM Thirteenth Conference on Information and Knowledge Management (CIKM 04), S. 553-561.

[Schaefer 2005]

Schaefer, André (2005): Integrierte Suche in heterogenen digitalen Bibliotheken mit Daffodil. Vortrag auf dem 4. Hildesheimer Evaluierungs- und Retrieval-Workshop (HIER) 2005.
http://www.is.informatik.uni-duisburg.de/bib/pdf/ir/Schaefer:05ta.pdf
(verifiziert am 28.03.2006)

[Shakes et al. 1997]

Shakes, Jonathan; Langheinrich, Marc; Etzioni, Oren (1997): Dynamic Reference Sifting: A case study in the Homepage Domain. In: Proceedings of the Sixth International World Wide Web Conference 1997, S.189-200.

[Sherman, Price 2001]

Sherman, Chris; Price, Gary (2001): The Invisible Web: uncovering information sources search engines can't see. Medford, NJ: CyberAge Books.

[Steele 2001]

Steele, Robert (2001): Techniques for Specialized Search Engines. In: Proceedings of the 2nd International Conference on Internet Computing (IC 2001).

[Wang, Hu 2002]

Wang, Yulin; Hu, Jianying (2002): Detecting Tables in HTML-Documents. In: Document Analysis Systems V. Proceedings of the 5th International Workshop, DAS 2002. New York et al.: Springer, S. 249-260.
[Lecture Notes in Computer Science, Vol. 2423]

[Xi et al. 2002]

Xi, Wensi; Fox, Edward A.; Tan, Roy P.; Shu, Jiang (2002): Machine Learning Approach for Homepage Finding Task. In: Laender, Alberto; Oliveira, Arlindo (Hrsg.) (2002): Proceedings of 9th international symposium on String processing and information retrieval (SPIRE 2002). Berlin et al.: Springer, S. 145-159.

Abbildungsverzeichnis

Anhang

Inhalt der CD

Auf der beiliegenden CD befinden sich folgende Inhalte:

- die vorliegende Arbeit im PDF-Format
- der Quellcode der Java-Applikation
- die JavaDoc-Dokumentation des Programms

Anhang

Anhang I: Trainingsmenge

Anhang II: Eigenschaften der Trainingsmenge

Anhang III: Testmenge

Anhang IV: Ergebnisse der Evaluierung der Informationsextraktion aus den Homepages

Anhang I: Trainingsmenge

(Auswahl der Autoren aus:

Eibl, M.; Wolff, Ch.; Womser-Hacker, Ch. (Hrsg.) (2005): Designing Information Systems. Festschrift für Jürgen Krause zum 60. Geburtstag. UVK: Konstanz)

(Erhebungszeitraum: Oktober 2005)

Vorname	Name	Homepage 1	Homepage 2
Maximilian	Eibl	http://www.tu-chemnitz.de/informatik/HomePages/Medieninformatik/prof/prof.php	http://www.gesis.org/GESIS_Aussenstelle/Eibl/
Norbert	Fuhr	http://www.is.informatik.uni-duisburg.de/staff/fuhr.html	
Rainer	Hammwöhner	http://www-iw.uni-regensburg.de/mamboiw/index.php?option=com_content&task=view&id=55&Itemid=85	
Ludwig	Hitzenberger	http://www-iw.uni-regensburg.de/mamboiw/index.php?option=com_content&task=view&id=101&Itemid=147	
Gerhard	Knorz	http://www.k-n-o-r-z.de/pers/knorz2.htm	
Jürgen	Krause	http://www.uni-koblenz.de/FB4/People/Person/krause	http://www.gesis.org/IZ/Krause/
Rainer	Kuhlen	http://www.kuhlen.name/	
Thomas	Mandl	http://www.uni-hildesheim.de/~mandl	
Rainhard	Oppermann	http://www.fit.fraunhofer.de/~oppi/	
Wolf	Rauch	http://www.kfunigraz.ac.at/iwiwww/pers/rauch.html	
Harald	Reiterer	http://hci.uni-konstanz.de/index.php?a=staff&b=Reiterer&c=contact&lang=de	
Maximilian	Stempfhuber	http://www.gesis.org/IZ/Stempfhuber/	
Christian	Wolff	http://www-cgi.uni-regensburg.de/cgi-bin/Medieninformatik/index.php?option=com_content&task=view&id=55&Itemid=85	
Christa	Womser-Hacker	http://www.uni-hildesheim.de/de/womser.htm	

Anhang II: Eigenschaften der Trainingsmenge

(Erhebungszeitraum: Oktober 2005)

HP	Name in Titel	akademischer Titel in Titel	Größe URL in kB	special feature in URL	Begriffe in Titel oder Snippet
1	J	J	22	J, uni	homepage
2	J	J	19	J, uni, informatik	informatik
3	J	J	9	N	
4	N	N	17	J	
5	J	J	12	J,uni, people, person	
6	J	J	15	J, uni, iw	
7	J	N	15	J, tu, informatik, homepage	informatik
8	J	J	26	N	informationswissenschaft
9	J	J	73	J, uni, staff	
10	J	J	16	J, name	informationswissenschaft
11	J	J	3	J, uni, iw	informationswissenschaft
12	J	N	5	J, ~	
13	J	N	16	Ja, uni, ~	homepage
14	J	N	16	J, iw	informationswissenschaft

(J: vorhanden, N: nicht vorhanden)

Anhang III: Testmenge

(Auswahl der Autoren aus:

Bernard Bekavac, Josef Herget, Marc Rittberger (Hrsg.): Information zwischen Kultur und Marktwirtschaft - Proceedings des 9. Internationalen Symposiums für Informationswissenschaft (ISI 2004), Chur, 6. – 8. Oktober 2004. Hochschulverband für Informationswissenschaft.)

(Erhebungszeitraum: März 2006)

Vorname	Nachname	HP
Rafael	Ball	http://www.fz-juelich.de/zb/index.php?index=52
Oliver	Bendel	http://web.iwi.unisg.ch/org/iwi/iwi_web.nsf/wwwTeamGer/BendelOliver.htm
Jochen	Brüning	http://www.inf-wiss.uni-konstanz.de/People/JB/
Alexander	Eckl	http://www2.informatik.uni-wuerzburg.de/staff/eckl/?i2statuslang=de
Hans W.	Giessen	http://vili.is.uni-sb.de/person_info.php?id=304
Juan	Gorraiz	http://www2.uibk.ac.at/ub/lis/juan_gorraiz.html
Joachim	Griesbaum	http://www.inf-wiss.uni-konstanz.de/People/jg.html
Josef	Herget	http://www.herget.ch/
Sonja	Hierl	http://www.fh-htwchur.ch/studien/diplomstudien/information_und_dokumentation/unser_team/?pvid=339
Michael	Kluck	http://www.swp-berlin.org/forscher/forscherprofil.php?id=4464
Ralph	Kölle	http://www.uni-hildesheim.de/koelle/
Marcello	L'Abbate	http://www.ipsi.fraunhofer.de/~labbate/
Norbert	Lang	http://www.fhnon.de/u/lang/www/index2.htm
Dirk	Lewandowski	http://www.phil-fak.uni-duesseldorf.de/infowiss/content/mitarbeiter/lewandowski.php
Dirk	Lewandowski	http://www.durchdenken.de/lewandowski/publikationen.php
Elisabeth	Milchrahm	http://www.kfunigraz.ac.at/iwiwww/pers/mil.html
Peter	Mutschke	http://www.gesis.org/IZ/Mutschke/
Daniel	Nerlich	http://www.ifg.ethz.ch/people/data/nerlich/index
Jana	Neuhaus	http://wwwcs.upb.de/cs/ag-szwillus/personen/mini/index.html
Daniel	Osterwalder	http://www.nhfi.de/mitglieder.php?ID=102
Annette	Pattloch	http://www.tfh-berlin.de/~pattloch/
Christian	Schlögl	http://www.kfunigraz.ac.at/iwiwww/pers/schl.html
Wolfgang	Semar	http://www.inf-wiss.uni-konstanz.de/People/ws.html
Robert	Strötgen	http://www.uni-hildesheim.de/~stroetge/

Anhang IV: Ergebnisse der Evaluierung der Informationsextraktion aus den Homepages - Teil 1 (Erhebungszeitraum: März 2006)

HP	akademischer Titel	E-Mail-Adresse	Bild
1	n.e.	n.e., @-Zeichen als Grafik,	n.a.
2	R	n.e., e-Mail-Adresse kein Verweis	n.a.
3	F, nicht vollständig extrahiert	R	R
4	R	n.e., Javascript	n.a.
5	n.e.	R	n.a.
6	R	n.e., Extra Kontaktseite	R
7	F, da falsch angegeben	n.e., kein Verweis, sondern Link zu Kontaktformular	n.e., Dateiname enthält Teil des Nachnamens
8	F, nicht vollständig extrahiert	R	n.e., Bilddatei nur mit Initialien bezeichnet
9	R	R	n.a.
10	F, anderer Titel extrahiert	R	R
11	n.a.	R	n.e., Bilddatei mit Initialen bezeichnet
12	n.a., F, anderer Titel extrahiert	R	n.e., Bilddatei nur mit Vorname benannt
13	R	R	F, Bilddatei enthält die Anfangsbuchstaben
14	R	R	R
15	n.a.	R	F, andere Grafik enthält den Namen
16	R	R	n.e., Name nicht in Dateiname
17	n.e., nicht gefunden, in Grafik enthalten	n.e., da 2 verknüpfte E-Mail-Adressen	n.e., Bilddatei mit Initialen bezeichnet
18	R	F, angegebene E-Mail-Adresse enthält nicht den Namen	R
19	n.a.	R	n.e., Name nicht in Dateiname, aber Portrait
20	R	R	R
21	F, nicht vollständig	R	n.e., Dateiname enthält Teil des Nachnamens
22	R	n.e., da auf Extra Kontaktseite	n.e., Name nicht in Dateiname
23	R	R	R
24	n.a.	n.e., E-Mail-Adresse als Grafik, PHP-Skript	R
25	R	R	n.e., Bilddatei mit Initialen bezeichnet
26	R	R	R

R: korrekt extrahiert, F, falsches Informationsobjekt extrahiert, n.e.: nicht gefunden

Anhang IV: Ergebnisse der Evaluierung der Informationsextraktion aus den Homepages II - Teil 2 (Erhebungszeitraum: März 2006)

HP	Publikationen	Projekte	Lebenslauf
1	R	R	n.a.
2	n.a.	n.a.	n.a.
3	R	n.a.	n.e., ohne Überschrift am Anfang der Seite
4	n.a.	n.a.	n.a.
5	n.a.	n.a.	n.a.
6	R	n.a.	n.a.
7	n.a.	n.a.	n.a.
8	R	n.a.	n.e., ohne Überschrift am Anfang der Seite
9	n.a.	n.a.	n.a.
10	F	F	n.a.
11	R	n.a.	R
12	n.a.	n.a.	n.a.
13	R,	R	n.e., KW Portrait
14	R	n.a.	n.e., KW Kurzvita
15	n.a.		
16	R	n.a.	n.e., ohne Überschrift am Anfang der Seite
17	n.a.	n.a.	F, KW cv
18	F, Fachbereichspublikationen	n.e.	n.e., ohne Überschrift am Anfang der Seite
19	R	R	n.a.
20	n.e., wird nicht als Label erkannt	n.a.	n.a.
21	F, Fachbereichspublikationen	F, Fachbereichsprojekte	n.e., KW "Person"
22	n.e., da Links als Image	n.a.	n.a.
23	n.e., Überschrift nicht auf Unterseite	R	n.a.
24	n.a.	n.a.	n.e., ohne Überschrift am Anfang der Seite
25	n.e., Seite auf Englisch	n.e.	Link defekt
26	R	n.a.	n.a.

R: korrekt extrahiert, F, falsches Informationsobjekt extrahiert, n.e.: nicht gefunden

Eigenständigkeitserklärung

Hier mit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Außerdem versichere ich, dass die Arbeit noch nicht veröffentlicht oder in einem anderen Prüfungsverfahren als Prüfungsleistung vorgelegt wurde.

Hildesheim, im März 2006

_____